

OPINNÄYTETYÖ

Jussi Tuoma 2010

WORDPRESS-BLOGIALUSTAN KÄYTTÖ WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄNÄ



Rovaniemen
ammattikorkeakoulu
University of Applied Sciences

TIETOJENKÄSITTELYN KOULUTUSOHJELMA

ROVANIEMEN AMMATTIKORKEAKOULU

LUONNONTIETEIDEN ALA

Tietojenkäsittelyn koulutusohjelma

Opinnäytetyö

WORDPRESS-BLOGIALUSTAN KÄYTTÖ WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄNÄ

Jussi Tuoma

2010

Toimeksiantaja Mainostoimisto Seven-1

Ohjaaja Aarre Jortikka

Hyväksytty _____ 2010 _____

Tekijä	Jussi Tuoma	2010
Toimeksiantaja	Mainostoimisto Seven-1	
Työn nimi	WordPress-blogialustan käyttö WWW-sisällönhallintajärjestelmänä	
Sivu- ja liitemäärä	39 + 11	

Opinnäytetyön aiheena oli selvittää WordPress-blogialustan käyttöä WWW-sivun sisällönhallintajärjestelmänä ja rakentaa WordPressillä mallisivusto, jossa on pyritty ottamaan huomioon WWW-sivun hallinta kokonaisvaltaisesti. Työhön sisältyi WordPressin taustan kartoittaminen, WordPress-sivuston rakentamiseen liittyvien vaiheiden läpikäyminen ja yleinen selvitys WordPress-sivupohjien ja lisäosien käytöstä. Opinnäytetyön toimeksiantaja oli mainostoimisto Seven-1, jonka toimipaikka sijaitsee Rovaniemellä.

Mallisivuston ulkoasulle ei tässä tapauksessa annettu kovin suurta painoarvoa, vaan tarkoituksena oli saada nimenomaan kokonaisvaltaisesti hyvin toimiva ja helposti hallittava sivusto. Mallisivustoa on tarkoitus käyttää jatkossa pohjana muiden projektien yhteydessä. Sivustoa varten luotiin myös kattava dokumentointi työn vaiheista ja kunkin sivuston elementin toiminnasta.

Työssä ei lähdetty erikseen vertailemaan ja valitsemaan sopivaa sisällönhallintajärjestelmää, koska toimeksiantaja oli jo aiemmin valinnut alustakseen WordPressin. Työssä käydään kuitenkin lyhyesti läpi myös muita markkinoiden johtavia avoimen lähdekoodin WWW-sisällönhallintajärjestelmiä ja niiden ominaisuuksia.

Työ oli kaiken kaikkiaan onnistunut projekti, ja tulee varmasti auttamaan toimeksiantajan uusien WWW-projektien läpiviennissä. Aikataulullisesti opinnäytetyöprojekti oli tiukka, johtuen toimeksiannon ajankohdasta. Mahdollisuus työn tekemiseen ilmeni vasta helmikuun loppupuolella.

Author	Jussi Tuoma	Year	2010
Commissioned by	Mainostoimisto Seven-1		
Subject of thesis	Use of a WordPress blog tool as a Content Management System		
Number of pages	39 + 11		

The purpose of this thesis was to solve how to use a WordPress blog tool as a Content Management System (CMS) for websites. The outcome of this work was a demo website that was running on WordPress. The website had to be built in a full and comprehensive CMS way to support this thesis. The thesis included the following workflow: the history of WordPress, stages of building a WordPress website and an overall report from the use of WordPress themes and plug-ins. This thesis was made as a commission from Seven-1 advertising agency based in Rovaniemi, Finland.

The purpose of the demo website was to build as consistent and comprehensive a website as possible and included the need for simple administration duties. The design of the layout was intentionally held back as a priority. As the demo website is going to be used in future projects at the Seven-1 company, a document from different stages of the work was created.

This thesis does not go deeply into comparing different CMS because the Seven-1 company had already made a decision to use WordPress as their main CMS for customers. However this thesis briefly discusses some key features of other leading open source Content Management Systems.

In the end the whole project was a success and it will be extremely useful to the Seven-1 company in the future when they are building new web-projects. The schedule for this project was very strict. The reason for this kind of schedule was the late date of the commission as the opportunity to undertake this project was only given in late February.

Key words

WordPress, CMS, Content Management System

Special remarks

SISÄLTÖ

1 JOHDANTO.....	2
2 PERUSKÄSITTEITÄ	3
2.1 PHP.....	3
2.2 MySQL	4
2.3 HTML/XHTML	4
2.4 CSS.....	5
3 WWW- SISÄLLÖNHALLINTAJÄRJESTELMÄ	8
3.1 WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄN MÄÄRITTELY	8
3.2 YKSITTÄISTEN HTML/XHTML-SIVUJEN KÄYTTÖTARKOITUS	10
3.3 MILLOIN YRITYKSEN TULISI OTTAA WWW-SISÄLLÖNHALLINTAJÄRJESTELMÄ KÄYTTÖÖN.....	11
4 WORDPRESS-SISÄLLÖNHALLINTAOHJELMA.....	13
4.1 WORDPRESSIN KEHITYSKULKU	13
4.2 WORDPRESS VERRATTUNA MUIHIN WWW-SISÄLLÖNHALLINTAJÄRJESTELMIIN	14
4.3 WORDPRESSIN HALLINTA JA KÄYTTÖLIITTYMÄ	17
5 WORDPRESS – SIVUSTON SUUNNITTELU JA TOTEUTUS.....	21
5.1 WORDPRESSIN SIVURAKENNE	21
5.1.1 WordPressin teemahierarkia	22
5.1.2 Sivupohjan rakenteen suunnittelu	23
5.2 SIVUPOHJAN RAKENTAMINEN JA ULKOASUN MUOKKAAMINEN.....	24
5.3 WORDPRESS-LISÄOSAT.....	28
5.3.1 Tekstin muokkaus ja lomakkeet.....	28
5.3.2 Sisällönhallintalisäosat	30
5.3.3 Hakukoneoptimointi	31
6 MALLISIVUSTON TOTEUTUS.....	33
6.1 LÄHTÖKOHDAT	33
6.2 SIVUSTON TOTEUTUS	33
6.3 KÄYTTÖTARKOITUS	34
7 YHTEENVETO	35
LÄHTEET	37
LIITTEET	40

1 JOHDANTO

Alun perin työni aiheena oli Joomla!-julkaisujärjestelmän käyttö sivuston ylläpitoon. Tämä tuntui luontevalta, koska olin rakentanut useita sivustoja kyseisellä julkaisujärjestelmällä, lähinnä pienyrityksille ja omiin tarpeisiini. Syksyllä 2009 aloin kuitenkin työskentelemään kokopäiväisesti nykyiselle toimeksiantajalleni, Mainostoimisto Seven-1:lle, jolloin suunnitelmani muuttuivat.

Aloittaessani Seven-1:issa, yhtiö oli vastikään alkanut käyttämään WordPress WWW-sisällönhallintaohjelmaa tekemiensä sivustojen ylläpitoon. Lähes ensimmäinen toimeksiantoni olikin tutustua kyseiseen ohjelmaan ja opetella tekemään sillä sivustoja. Pikkuhiljaa ajatus opinnäytetyöstä ja siihen liittyvästä sivuston rakentamisesta Joomla:n päälle kuihtui, koska en halunnut enää palata kyseisen WWW-julkaisujärjestelmän pariin.

Pyöriteltyni WordPress-sivustoja noin 5 kuukauden ajan Seven-1:n työntekijänä, tuli yhtiöltä ehdotus, että tekisin opinnäytetyöni heille ja päätin tarttua tilaisuuteen. Helmikuussa 2010 olimme tehneet Seven-1:issa jo useita sivustoja WordPressillä, mutta kunnollinen dokumentointi ja yhteenveto siitä, mitä kaikkea kyseisellä WWW-julkaisuohjelmalla pystyy tekemään loisti yhä poissaolollaan. Yhtiö tarvitsi selkeän mallisivuston, johon liittyi hyvä dokumentointi. Sivuston, joka toimisi selvästi lähtökohtana muille sivustoille, ja jota olisi hyvä soveltaa uusiin projekteihin.

Itse työn on tarkoitus selventää miten WordPress –ohjelmaa, joka on suunniteltu blogialustaksi, voidaan käyttää WWW-sivustojen sisällönhallintaohjelmanä. Työ toivottavasti selventää lukijalle, millaisia vaiheita sivuston rakentamiseen kuuluu ja mitä kaikkea tulee ottaa huomioon, kun se tehdään WordPress-sisällönhallintaohjelman päälle. Opinnäytetyössäni on poissuljettu eri WWW-sisällönhallintajärjestelmien vertailu ja niiden perusteella tehty sisällönhallintajärjestelmän valinta, koska yhtiö oli valinnut jo aiemmin käyttöönsä WordPressin.

2 PERUSKÄSITTEITÄ

Luvun tarkoitus on kuvailla opinnäytetyöhön liittyvät peruskäsitteet ja syy miksi ne ovat tärkeä osa opinnäytetyötä.

2.1 PHP

PHP on lyhenne sanoista PHP: Hypertext Preprocessor. PHP:tä käytetään erityisesti WWW-palvelinympäristöissä dynaamisten WWW-sivujen luomiseen. PHP on komentosarjakieli, eli skriptikieli, jossa ohjelmakoodi tulkitaan vastan ohjelman suoritussvaiheessa. Kielenä PHP on alustariippumaton, eli se ei ole sidoksissa tiettyyn laitteistoalustaan tai käyttöjärjestelmään. PHP-koodia käytetään yleensä HTML/XHTML-koodiin upotettuna luomaan dynaamista sisältöä sivustolle. (PHP: Manual 2010.)

WordPress, kuten suurin osa markkinoilla olevista avoimen lähdekoodin julkaisujärjestelmistä perustuu PHP-koodiin. Tämä käy ilmi muun muassa OpenJason- ja CMS Critic –sivustojen sisällönhallintajärjestelmien listauksista (OpenJason: 50 Content Management Systems 2010, CMS Critic 2010) sekä Wikipedian eri sisällönhallintajärjestelmien taulukkomaisesta listauksesta, jossa on listattuna yli 90 avoimen lähdekoodin sisällönhallintaohjelmaa (Wikipedia 2010b). Näistä noin puolet perustuu PHP-koodiin. Syy PHP-julkaisujärjestelmien selvästi suurempaan määrään markkinoilla on sen yksinkertainen ja helposti omaksuttavissa oleva syntaksi, jolloin myös kehittäjiä on enemmän. Muita syitä ovat PHP-kielen vakaus, PHP-ympäristön edullinen hostaus ja kielen alustariippumattomuus (2K Mediat 2010). Hostauksella tarkoitetaan webhotellipalvelua (engl. web hosting), jossa palvelintila ja siihen liittyvien palvelinohjelmistojen ajaminen annetaan jonkin toisen yrityksen hoidettavaksi. Webhotellien hinnat määräytyvät webhotellin ominaisuuksien mukaan ja edullisimpiin paketteihin kuuluu yleensä lähes poikkeuksetta tuki PHP:lle (Webhotellien vertailu 2010).

2.2 MySQL

MySQL on SQL-tietokannan (engl. Structured Query Language), eli relaatiotietokannan hallintajärjestelmä. SQL-tietokanta mahdollistaa erilaisten kyselyjen eli tietokantahakujen luomisen esimerkiksi PHP-kielellä. MySQL on avoimen lähdekoodin tietokantajärjestelmä ja se on asennettu jo yli kuuteen miljoonaan tietokoneeseen. (MySQL 2010.)

MySQL mahdollistaa WWW-sisällönhallintajärjestelmän toiminnan. Myös muut relaatiotietokannat, kuten PostgreSQL and SQLite voivat toimia WWW-sisällönhallintajärjestelmän taustalla, mutta WordPress on valinnut tietokantajärjestelmäkseen MySQL:n. Suurimmat syyt valinnan taustalla ovat MySQL:n nopeus, yleisyys ja tietysti se, että se on ilmainen.

Vuonna 2009 Oracle Corporation ilmoitti ostaneensa Sun Microsystemsin, joka sai WordPressin käyttäjät huolestumaan. Sun Microsystems oli MySQL tietokantajärjestelmän omistaja ja Oracle yksi sen pahimpia kilpailijoita omalla Oracle 11g –tietokantajärjestelmällään. WordPressin käyttäjiä huolestutti luonnollisesti MySQL järjestelmän tulevaisuus ja sitä myöten WordPressin tulevaisuus ilmaisena ohjelmana. Oracle on kuitenkin vakuuttanut, että MySQL tulee olemaan osa Oraclen tarjoamia tuotteita ja sen kehitystyön luvataan jatkuvan (Oracle 2010). Tämän lisäksi yksi WordPressin perustajista ja pääkehittäjistä Matt Mullenweg kertoi blogissaan luottavaisin mielin MySQL:n kehitystyön jatkuvan ja väläytti mahdollisuutta siitä, että WordPressiin tuotaisiin tuki muillekin relaatiotietokannoille. Suunnitelmissa on rakentaa tuki muille tietokannoille siten, että se ei vaatisi käyttäjiltä sivupohjien tai lisäosien uudelleen koodaamista. Tällöin huoli MySQL:n tulevaisuudesta olisi poissuljettu ainakin hieman pidemmäksi aikaa. (Matt Mullenweg 2010).

2.3 HTML/XHTML

HTML on lyhenne sanoista Hypertext Markup Language ja XHTML on lyhenne sanoista eXtensible Hypertext Markup Language. Molemmat ovat WWW-kuvauskieliä, jotka soveltuvat WWW-sivujen rakentamiseen. XHTML-kieli

kehitettiin korvaamaan HTML-kieli, ja yksi sen tavoitteista oli saada WWW-kuvauskieli toimimaan paremmin muissakin laitteissa kuin pelkästään tietokoneissa. XHTML-kieli ei onnistunut kuitenkaan korvaamaan HTML kieltä, vaan vuonna 2007 HTML-kielen kehitys aloitettiin uudelleen. Tällä hetkellä sekä HTML- että XHTML-kielistä on tulossa uudet versiot (HTML5 ja XHTML5) W3C:n (World Wide Web Consortium) mukaan vuoden 2010 loppupuolella. Uudet versiot muuttavat tällä hetkellä olevaa kuvauskielen rakennetta merkittävästi ja tulevat helpottamaan sivujen rakentamista. Uusien ominaisuuksien tukeminen kaikkien merkittävien selainvalmistajien puolelta voi kylläkin kestää useita vuosia, pahimmissa arvioissa jopa 12 vuotta, joten ihan heti uusista ominaisuuksista ei päästä nauttimaan (Webmonkey.com 2008). (W3C 2010b, XHTML™ 1.0 The Extensible HyperText Markup Language 2010 (Second Edition).)

XHTML/HTML-kieli on oleellinen osa WordPressiä ja muodostaa rungon, jonka sisään sisältö tuodaan tietokannasta. Voidakseen luoda näyttäviä sivupohjia tai muokata jo olemassa olevia sivupohjia, on sivuston ylläpitäjän osattava XHTML/HTML-kieltä. Sivuston ulkoasun muokkaamisen kannalta tämä on jopa tärkeämpi taito kuin PHP-osaaminen.

2.4 CSS

CSS on lyhenne sanoista Cascading Style Sheets, se on yksinkertainen tyylikieli, jonka avulla määritellään HTML, XHTML ja XML -dokumenttien ulkoasu ja esitystapa. Tällä hetkellä on käytössä CSS 2.1, jota kaikki nykypäivän selaimet tukevat ainakin rajallisesti. Johtavien selainvalmistajien CSS-tyylistandardien täydellisen tuen puute on jo useiden vuosien ajan aiheuttanut päänvaivaa, ylimääräisiä kustannuksia ja aikaa vievää testaamista WWW-sivujen kehittäjien keskuudessa. (W3C 2010a.)

Yleisesti tiedossa ovat esimerkiksi Microsoftin Internet Explorer -selainten ongelmat lukea CSS:ää oikein. Pahimmat ongelmat koskevat vuonna 2001 julkaistua Internet Explorerin versiota 6, jonka käyttäjämäärät ovat onneksi laskeneet viimeisten vuosien aikana suuresti. Version 7 piti ilmestyessään

vuoden 2006 lokakuussa korjata suurimmat ongelmat, mutta selain oli alusta alkaen epäonnistunut. Microsoft julkaisikin seuraavan version selaimestaan jo vuoden 2009 maaliskuussa, joka on osoittautunut myös huonoksi CSS:n web standardien tukemisen suhteen. (Wikipedia 2010a.)

Internet Explorerin ongelmat on helppo todeta Acid2- ja Acid3-selaintesteillä. Testit on suunniteltu tarkistamaan selaimien ominaisuuksia ja kykyä lukea WWW-sivuja oikein. Testit perustuvat W3C:n määrittämiin web standardeihin. Acid2-testi on tarkoitettu HTML4-kielen ja CSS 2.1 –standardien testaamiseen, ja testissä on otettu huomioon perusominaisuudet, jotka modernien selainten pitäisi osata tulkita. Perusominaisuuksia ovat muun muassa läpinäkyvät PNG-kuvat, object-elementit, CSS-jäsentely, marginaalien laskeminen ja runkoelementtien, eli div-tagien asettelu ja käyttäytyminen. Acid3-testi on tarkoitettu lähinnä testaamaan selainten valmiuksia ajaa nykyaikaisia dynaamisia websovelluksia sekä sitä, kuinka hyvin ne tukevat CSS3-versiota. (Webstandards.org 2010.)

Tein itse Acid2- ja Acid3-testit Mozilla Firefox selaimen versioilla 2.0 ja 3.6; Internet Explorerin versioilla 6, 7 ja 8 sekä Operan ja Safarin uusimmilla versioilla. Vain Operan ja Safarin selaimet läpäisivät molemmat testit. Mozilla Firefox 3.6 läpäisi Acid2-testin ja sai Acid3-testissä 94/100, mutta vanhempi versio 2.0 ei läpäissyt kumpaakaan testiä. Internet Explorerin versioista yksikään ei läpäissyt kumpaakaan testiä. Testitulokset ovat nähtävissä tarkemmin liitteessä 2. (Webstandards.org 2010.)

Internet Explorer on menettänytkin viimeisten vuosien aikana todella paljon käyttäjiä, joka käy ilmi kansainvälisestä selaintilastosta. StatCounter on verkossa toimiva ilmainen palvelu, joka listaa tilastoja yli 3 miljoonalta verkkosivulta maailmanlaajuisesti (3 miljoonan verkkosivun raja ylittyi elokuussa 2009). Tilastot päivitetään kerran kuukaudessa ja tilastoista käy ilmi muun muassa se, että Suomessa yleisin selain on ollut jo heinäkuusta 2008 lähtien Mozilla Firefox. Tilastoista on nähtävissä myös Internet Explorer -selaimen käyttäjämäärien selvä

lasku kaikkialla maailmassa ja muiden selainten selvä nousujohteinen trendi. Selaintilastot ovat listattuna tarkemmin liitteessä 3. (StatCounter 2010.)

W3C:llä on tällä hetkellä kehitteillä CSS3 –versio, jonka ominaisuuksiin kuuluu muun muassa divien pyöristetyt reunat, divien varjot, tekstivarjot, liukuvärit ja monia muita ominaisuuksia, joita ei ole voinut toteuttaa aikaisemmin kuin kuvien avulla. CSS3:sta voidaan kuitenkin käyttää tehokkaasti hyväksi vasta kun kaikki suurimmat selainvalmistajat tukevat tekniikkaa tarpeeksi hyvin. (W3C Cascading Style Sheets 2010.)

CSS on WordPress-sivustojen kannalta todella tärkeä, koska kaikkia WordPressin sivupohjien ulkoasuun liittyviä elementtejä voidaan määritellä haluamukseen CSS-tyylitiedostojen avulla.

3 WWW- SISÄLLÖNHALLINTAJÄRJESTELMÄ

Tässä luvussa käydään lyhyesti läpi WWW-sisältöhallintajärjestelmän määritelmät. Milloin julkaisualusta täyttää WWW-sisällönhallintajärjestelmän vaatimukset. Mikä on sisällönhallintajärjestelmän tarkoitus? Lisäksi tarkastellaan tietokantapohjaisten WWW-järjestelmien etuja ja vaatimuksia suhteessa joukkoon yksittäisiä, perinteisiä HTML/XHTML-sivuja. Millaiseen käyttötarkoitukseen perinteiset HTML/XHTML-sivut soveltuvat paremmin kuin julkaisujärjestelmät ja mitä voidaan pitää ratkaisevana käännepisteenä, jonka jälkeen yritys tarvitsee tietokantapohjaista julkaisujärjestelmää?

3.1 WWW-sisällönhallintajärjestelmän määrittely

Tässä työssä on tietoisesti rajattu sisällönhallinta koskemaan pelkästään WWW-sisällönhallintaa, koska sisällönhallinta on terminä paljon laajempi mitä yleensä oletetaan. Bob Boikon mukaan sisällönhallinta tarkoittaa lähemmässä tarkastelussa kokonaisvaltaista prosessia, joka sisältää sisällön keräämisen, hallinnan ja julkaisemisen. Sisällönhallinnan tarkempi käsittely olisi jo oman opinnäytetyön laajuinen projekti, joten tässä opinnäytetyössä sitä ei käsitellä. Tässä opinnäytetyössä käydään kuitenkin lyhyesti läpi WWW-sisällönhallintajärjestelmien määrittelyt. (Boiko 2005, 72-74.)

Boiko on jakanut WWW-sisällönhallintajärjestelmät viiteen kategoriaan. Nimellinen WWW-sisällönhallintajärjestelmä (engl. The nominal Web CMS) on Microsoft FrontPagen tai Adobe Dreamweaverin (ent. Macromedia Dreamweaver) kaltainen ohjelma, jonka avulla voidaan luoda sisältöä What You See Is What You Get (WYSIWYG)-menetelmällä, jonka tarkoitus on näyttää sivu ohjelmassa juuri sellaisena kuin se tulee näkymään WWW-sivuilla. Nämä ohjelmat mahdollistavat myös sivupohjien rakentamisen ja niiden jakamisen usealle ylläpitäjälle, mikä mahdollistaa yhtenäisen sivuston rakentamisen. Tällaisten ohjelmien vahvuuksia ovat myös tiedostokirjastot kuville ja muulle mediasisällölle, sekä linkkien ja sivujen systemaattinen hallinta. Nimellinen www-

sisällönhallintajärjestelmä alkaa olemaan jo kuollut käsite, mutta niin kauan kuin sitä tukevia ohjelmia riittää, se on olemassa. (Boiko 2005, 72-74.)

Dynaamisen WWW-sivut toimivat periaatteeltaan samalla tavoin kuin WWW-sisällönhallintajärjestelmät, eli ne hakevat esimerkiksi PHP-koodin avulla suoraan tietokannasta sisällön ja upottavat sen valmiiseen sivupohjaan. Kun käyttäjältä on lähtenyt pyyntö näyttää jokin sivu, prosessoidaan pyyntö palvelimella ja näytetään prosessoitu sivu käyttäjän internet selaimessa. Täsmälleen samalla tavoin toimii myös WWW-sisällönhallintajärjestelmä. Ero dynaamisten sivujen ja WWW-sisällönhallintajärjestelmän välillä on siinä, että sisällönhallintajärjestelmällä voidaan luoda myös staattisia sivuja. Aina ei ole tarvetta luoda pelkästään dynaamisia sivuja, jotka tekevät tietokantaan kyselyn aina kun selain päivitetään. Huomattavasti useammin luodaan sivuja, joiden sisältö on pysyvää. Tällaisten staattisten sivujen lataaminen on nopeampaa, koska sen ei tarvitse rasittaa palvelinta ylimääräisillä tietokantakyselyillä. Rajoittuneen rakenteensa lisäksi dynaamiset WWW-sivut ovat tarkoitettu vain WWW-sivujen luomiseen ja jos niille alkaa kehittämään muita elementtejä, kuten laajempaa ylläpitohallintaa tai varmuuskopiointimahdollisuutta, luodaan silloin jo itse asiassa erillistä WWW-sisällönhallintajärjestelmää. (Boiko 2005, 75-77.)

WWW-sisällönhallintajärjestelmä (engl. The Web CMS) sisältää sekä dynaamisia että staattisia sivuja ja kaikki on hallittavissa yhdellä ylläpitojärjestelmällä. Ylläpitojärjestelmä voi olla omalla koneella lokaalisti (lähiverkossa) tai se voi sijaita verkkopalvelimella. Kokonaisvaltainen WWW-sisällönhallintajärjestelmä sisältää ohjelman sisällön luomiseen sekä median-, sivupohjien ja käyttäjien hallintaan. (Boiko 2005, 77-79.)

Kokonaisvaltainen sisällönhallintajärjestelmä yhdistää sekä WWW-sisällönhallinnan että printtimedian ja sähköpostiliikenteen sisällönhallinnan. Kokonaisvaltaisen sisällönhallintajärjestelmän tarkoitus on edistää sisällöntuotannon yhtenäistämistä kaikkien mediatyyppien kesken. Se pakottaa yritykset luopumaan vanhanaikaisesta ajattelutavasta, jossa printtimedian ja

sähköisen median sisällöntuotanto on erotettu toisistaan. Kokonaisvaltaisessa sisällönhallinnassa pyritään siihen, että sisältö tuotetaan kertaalleen kaikille medioille sopivaksi. (Boiko 2005, 79-81.)

Yritykselle suunnattu sisällönhallintajärjestelmä (engl. The Enterprise CMS) eroaa kokonaisvaltaisesta sisällönhallintajärjestelmästä siinä, että sen avulla pyritään hallitsemaan mahdollisimman laajasti yhtiöön liittyvää informaatiota ja dataa. Tällainen ratkaisu käsittää sekä ulospäin suuntautuvan että yhtiön sisäisesti kulkevan sisällönhallinnan mahdollisimman kokonaisvaltaisesti. Ratkaisulla ei kuitenkaan voida hallita kaikkea yrityksen dataa, vaan sen tarkoitus on vähentää työvaiheita ja estää kertaalleen tehdyn työn toisto sekä helpottaa informaation käsittelyä ja löytämistä yrityksen sisällä. (Boiko 2005, 81-82.)

3.2 Yksittäisten HTML/XHTML-sivujen käyttötarkoitus

Tietokantapohjaisten WWW-sisällönhallintajärjestelmien historia juontaa juurensa 1990-luvun puolivälin tienoille, jolloin ensimmäiset kaupalliset ja patentoidut järjestelmät nousivat markkinoille. Tällaiset järjestelmät olivat kuitenkin silloin kalliita ja oikeastaan ennen vuosien 2001-2004 tapahtunutta avoimen lähdekoodin WWW-sisällönhallintajärjestelmien esiinmarssia oli yleisin tapa tehdä sivustot käsin koodaamalla tai jotakin sivujen tekoon tarkoitettua ohjelmaa, kuten Frontpagea tai Dreamweaveria käyttäen. (CMS Calender 2010.)

Kun mietitään yksittäisten HTML/XHTML-sivujen etuja ja haittoja, on niitä lähdeittävä purkamaan projektin luonteen näkökulmasta. Yksittäisillä HTML/XHTML-sivuilla tarkoitan nyt siis sivuja, jotka eivät käytä sisällönhallintajärjestelmiä tai yleensääkään mitään tietokantapohjaista ratkaisua tiedon esittämiseen WWW-sivuilla.

Nykyään yksittäisten HTML/XHTML -sivujen käyttö on perusteltua vain hyvin pienillä sivustoilla, jossa sivujen kokonaismäärä on esimerkiksi 10–20 ja muuttuvan sisällön määrä suhteessa pientä. Suuremmat sivustot ja varsinkin tiheästi päivitettävät sivustot tulisi ehdottomasti rakentaa jonkinlaisen WWW-sisällönhallintajärjestelmän päälle.

Pienet minisivustot (engl. mini-website) voi tehdä ilman minkäänlaista julkaisualustaa ja monesti se on ihan perusteltuakin. Jos tehdään esimerkiksi promootiotarkoituksessa jokin 3-6 sivun laajuinen sivusto, jonka sisältö on pääosin pysyvää, ei ole mitään ehdotonta tai edes järkevää syytä tehdä sivustoa WWW-julkaisualustan päälle. Minisivustojen hyviä käyttötarkoituksia ovat yleensä esimerkiksi tietyn tuotteen markkinointi tai uuden tuotteen testaaminen kohdeyleisöllä. Minisivustot voivat olla ulkoasultaan hyvinkin omalaatuisia, eikä niiden tarvitse välttämättä noudattaa perinteisen WWW-sivun logiikkaa käyttöliittymän tai ulkoasun osalta. Lisäksi markkinointiin ja promootiotarkoitukseen luodut sivustot ovat netissä vain sille tarkoitetun kampanjan ajan. (Mini Site Definition 2010.)

Yksittäiset HTML/XHTML-sivut eivät vaadi palvelimelta tietokanta- tai PHP-tukea ja ne ovat nopeammin rakennettavissa. Lisäksi voidaan miettiä asiaa taloudellisesta näkökulmasta, sillä vaikka nykyään webhotelli-palveluntarjoajien halvimpiinkin paketteihin kuuluu lähes poikkeuksetta PHP-tuki, ne eivät usein sisällä tietokantaa, vaan se on tulee vasta kalliimpien pakettien mukana tai se on mahdollisesti saatavissa erikseen lisäpalveluna. Lisäksi yksittäisten HTML/XHTML-sivujen etu sisällönhallintajärjestelmiin on nopeampi latausaika, koska selaimen ei tarvitse tehdä palvelimelle tietokantahakuja lainkaan, vaan tieto on luettavissa sivuilta suoraan (Stamatiou 2006). (Webhotellit24 2010, Webhotellien vertailu 2010.)

3.3 Milloin yrityksen tulisi ottaa WWW-sisällönhallintajärjestelmä käyttöön

Viimeisten 5-10 vuoden aikana kynnys WWW-sisällönhallintajärjestelmän hankkimiseen on laskenut huomattavasti. Sisällönhallintaohjelmia on nyt vuonna 2010 tarjolla enemmän kuin koskaan ja ne ovat erittäin edullisia, suuri osa jopa ilmaisia, jos niiden käyttöönottoon löytyy osaamista. Kun lisäksi palvelintilaa saa kattavilla ominaisuuksilla edullisesti, on suurimmat esteet WWW-sisällönhallintajärjestelmän käyttöönotolle ylitetty. Organisaatioiden sisällä ei ole enää samanlaista kartoittamisen tarvetta kalliin järjestelmän käyttöönotolle kuin mitä oli vuosituhanen alussa. WWW-sisällönhallintajärjestelmän käyttöönoton

tarve voidaan kuitenkin yhä määritellä lähes samalla tavoin kuin kymmenen vuotta sitten.

WWW-sisällönhallintajärjestelmän tarve heijastuu hyvin pitkälti organisaation WWW-sisällön luonteeseen. Jos organisaatiolla on tarvetta julkaista uutta informatiivista tietoa uutisten omaisesti säännöllisin väliajoin ja WWW-sisällön tuottamiseen osallistuu useita tahoja, voi tämä olla jo riittävä kynnys WWW-sisällönhallintajärjestelmään siirtymiselle. Huomioon otettavaa on kuitenkin organisaation tilanne. Jos organisaatio kokee, että sisällön tuottaminen ja sisältötekstien vedosvaihe, niiden siirteleminen työntekijältä toiselle on sujuvaa ja organisaation käytössä on joko sisäisesti tai alihankintana palkattu pätevä webmaster, ei tällainen tarve ole vielä ratkaiseva. (Rob Prideaux 2004.)

Tarve WWW-sisällönhallintajärjestelmälle voidaan katsoa ratkaisevaksi kun sisällön siirtäminen WWW-sivuille koetaan vaikeaksi tai kalliiksi prosessiksi. Syitä tällaiseen voivat olla mahdollisesti kommunikointivaikeudet alihankintana palkatun webmasterin tai mainostoimiston kanssa tai sisäisen vedosliikenteen takkuilu WWW-sisällöntuoton yhteydessä. Lisäksi jos päivityksen tarve WWW-sisällölle on viikoittaista ja organisaatiosta löytyy halukas tai päivittämiseen muuten sopiva työntekijä, mutta joka ei osaa HTML-koodikieltä, on WWW-sisällönhallintajärjestelmä hyvin perusteltu ja järkevä ratkaisu. WWW-sisällönhallintajärjestelmä auttaa vedosliikenteen vähentymiseen ratkaisevasti, kun kaikki työntekijät voivat käydä muokkaamassa samaa dokumenttia, mutta eri oikeuksilla. Kun dokumentti on saanut kaikkien hyväksynnän, voidaan se julkaista sille oikeudet omaavan työntekijän toimesta. Lisäksi tietokantaan tallentuvat dokumentin reviisiot, eli dokumentin muokkaushistoria, jota voidaan tutkailla suoraan järjestelmästä käsin. Tällainen ominaisuus löytyy ainakin WordPressistä oletuksena. (Rob Prideaux 2004.)

4 WORDPRESS-SISÄLLÖNHALLINTAOHJELMA

Tässä luvussa käydään läpi WordPressin taustaa. Mistä WordPress on saanut alkunsa, mihin tarkoitukseen se on luotu ja miksi se on päässyt niin hyvää asemaan avoimen lähdekoodin WWW-julkaisualustojen keskuudessa.

4.1 WordPressin kehityskulku

WordPress on henkilökohtainen julkaisualusta, joka on kehitetty alun perin blogien ylläpitoon. Se on avoimeen lähdekoodiin perustuva ja se on kirjoitettu PHP-ohjelmointikielellä käyttäen MySQL-tietokantahallintajärjestelmää tiedostojen tallentamiseen. WordPressin kehitys on ollut nopeaa ja vakuuttavaa, se on yksi suosituimmista WWW-sisällön hallintajärjestelmistä tällä hetkellä ja sitä voidaan käyttää sujuvasti suuremmankin WWW-sivuston sisällönhallintajärjestelmänä.

WordPress on saanut alkunsa Michael Valdrighi:in vuonna 2001 kehittämästä b2/cafelog:sta. B2/cafelog oli kirjoitettu PHP-ohjelmointikielellä, jossa blogikirjoitukset talletettiin MySQL-tietokantaan. B2/cafelog:in kehitys lopetettiin vuoden 2003 alkupuolella, jolloin sille oli kertynyt noin 2000 aktiivista käyttäjää. (Valdrighi 2003.)

B2/cafelog:in lopetuspäätöksen jälkeen Matt Mullenweg ja Mike Little haarauttivat (engl. fork) b2/cafelogin omaksi projektikseen ja alkoivat kehittää WordPressiä (Wheeler 2007). Micheal Valdrighi liittyi projektin kehitykseen mukaan ja ensimmäinen versio julkaistiin jo toukokuussa 2003. Heti ensimmäisestä versiosta (0.7) lähtien se oli täysin XHTML/CSS standardit täyttävä. Ylläpitopuolen suurimpia kantavia teemoja olivat heti alusta alkaen helppokäyttöisyys ja selkeys. (Valdrighi 2003.)

Vuonna 2004 julkaistiin versio 1.0, jonka yksi mielenkiintoisimmista ja parhaista uudistuksista oli kestoplinkkien (engl. permalinks) lisääminen. Kestolinkit muodostivat huomattavasti hakukoneystävällisemmän osoiterakenteen sivustolle verrattuna dynaamisesti luotuihin osoiterakenteisiin. (SEOmoz 2010). Muita

suuria uudistuksia versiossa 1.0 olivat kommenttien hallinta ja useiden kategorioiden käyttö blogikirjoituksissa. Versio 1.2 oli myös merkittävä, kun lisäosat (plugins) esiteltiin ensimmäistä kertaa. Vuonna 2005 julkaistujen versioiden 1.5 ja 2.0 myötä ohjelma otti suuren harppauksen kohti sisällönhallintajärjestelmää, jolloin se mahdollisti ”staattiset sivut”, eli pysyvän tiedon esittämisen ja kehittyneemmän template-järjestelmän. (WordPress.org 2010b.)

Version 2.0 jälkeen suurin yksittäinen versio on ollu 2.7, joka muun muassa uudisti koko hallintapaneelin, toi mukanaan lisäosien automaattisen asennuksen ja mahdollisti kommentteihin vastaamisen suoraan hallintapaneelin puolelta. Nämäkin uudistukset toivat mukanaan paljon uutta nimenomaan WWW-sisällönhallintaa ajatellen. (WordPress.org 2010b.)

Tällä hetkellä WordPress on valmistelemassa version 3.0 julkaisua, joka on edennyt beta 2 vaiheeseen. Versio 3 integroi ohjelmaansa WordPress MU:n, tuo mukanaan uuden navigoinnin rakennusosion, sekä muun muassa uudelleen rakennetun sisällön viemiseen ja tuomiseen (engl. export ja import) tarkoitetun työkalun. WordPress MU on useiden blogien ylläpitoon tarkoitettu hallintajärjestelmä, joka on tähän mennessä ollut erillinen ohjelmansa WordPressin tuoteperheessä. Versiosta 3 lähtien se tulee olemaan tämän hetken tiedoilla lisäosa WordPressiin. Kaikki suurimmat uudistukset koskien versiota 3 on selkeä askel kohti kokonaisvaltaisempaa sisällönhallintajärjestelmää, joka tulee yhdessä uusien lisäosien kanssa viemään WordPressin ihan uudelle tasolle. (WordPress.org 2010c.)

4.2 WordPress verrattuna muihin WWW-sisällönhallintajärjestelmiin

WordPressin laajempi vertaaminen muihin WWW-sisällönhallintajärjestelmiin olisi jälleen itsenäisen opinnäytetyön laajuinen projekti. Tässä työssä käydään kuitenkin lyhyesti läpi kahden muun avoimen lähdekoodin WWW-sisällönhallintajärjestelmän ominaisuuksia yleisellä tasolla ja verrataan niitä WordPressin ominaisuuksiin. Luvussa käydään läpi, miten nämä kolme WWW-

sisällönhallintajärjestelmää eroavat toisistaan ja onko kaikilla sama käyttötarkoitus, vai ovatko kaikki suunniteltu erilaisia WWW-toteutuksia silmälläpitäen.

Kaksi muuta WWW-sisällönhallintajärjestelmää WordPressin lisäksi ovat avoimeen lähdekoodiin perustuvat Joomla! ja Drupal. Nämä kolme ohjelmaa ovat suosituimmat avoimeen lähdekoodiin perustuvista WWW-sisällönhallintajärjestelmistä, joka käy ilmi muun muassa CMS Wire –sivuston vuonna 2009 teettämästä kyselystä. (CMS Wire 2009). Kaikilla näillä sisällönhallintaohjelmilla on laaja tukijoukko ja aktiivinen kehitystiimi takanaan. Tässä opinnäytetyössä käytän tästä eteenpäin Joomla!-sisällönhallintajärjestelmästä yksinkertaisemmin nimeä Joomla.

Kaikki kolme järjestelmää perustuvat PHP-kieleen ja ne käyttävät tietojensa tallentamiseen MySQL tietokantajärjestelmää. Nämä järjestelmät perustuvat samaan logiikkaan: PHP-skripteillä, eli ohjelmakohtaisilla PHP-tageilla ohjataan tietokantaa luomaan sivupohjien avulla sisältöä loppukäyttäjän WWW-selaimelle.

Suurimmat erot järjestelmien välillä tulee niiden käyttötarkoituksessa, eli mihin ne on alunperin luotu. Juuri tämän takia näiden suoranainen vertailu ei ole järkevää. WordPress on alunperin suunniteltu pienten blogimaisten sivustojen ylläpitoon, jonka tärkeimpiä ominaisuuksia ovat selvästi tekstin ja yleensäkin sisällön hallitseminen. Käyttöliittymältään WordPress on suunniteltu loppukäyttäjää ajatellen mahdollisimman selkeäksi ja yksinkertaiseksi. Tämän vuoksi sivuston kehittäjä näkee usein WordPressin ylläpitokäyttöliittymän vajavaiseksi kokonaisvaltaiseen sisällönhallintaan, mutta peruskäyttäjät kokevat sen juuri sopivaksi sisällöntuotantoa ajatellen.

Joomla ja Drupal ovat puolestaan suunniteltu enemmän kokonaisvaltaiseen sisällönhallintaan jo ensimmäisistä kehitysversiona alkaen, jolloin ne molemmat sisältävät enemmän säädettäviä osa-alueita ylläpitokäyttöliittymän puolelta (Liite 4). Tämän vuoksi molempien ylläpitokäyttöliittymä voi olla aluksi vaikeasti ymmärrettävissä kokemattomalle sisällönhallintaohjelman käyttäjälle.

Molemmista löytyy WordPressiä monipuolisempi käyttäjähallinta ja molempien ominaisuudet ovat kattavammat oletusasennuksen jälkeen.

WordPress on saavuttanut suosionsa kuitenkin sen vuoksi, että useimmat käyttäjät tarvitsevat omia WWW-sivujaan juuri blogin tai pienimuotoisen sivuston ylläpitoon, johon on helppo tuottaa ja muokata sisältöä WYSIWYG-periaatteella. Helppous ja nopeus ohjaavat nykyajan ihmistä lähes kaikissa toimissa. Osaavissa käsissä WordPress on taivuteltavissa todella näyttäviin ja toimiviin ratkaisuihin, mutta pitää muistaa sen rajoitteet, mihin se on alunperin suunniteltu. Joomla on helpommin omaksuttavissa käyttäjille, jotka eivät omaa koodaamistaitoja ja haluavat toimivan sisällönhallintajärjestelmän suoraan käyttövalmiina (engl. out of the box). Suuremmat rakenteelliset muutokset sivupohjiin vaativat kuitenkin koodaamistaitoja sekä paljon kärsivällisyyttä ja aikaa tutustua järjestelmän logiikkaan ja toimintaan. Drupal on Joomlaan tavoin suoraan käyttövalmis sisällönhallintajärjestelmä ja sisältää paljon ominaisuuksia valmiiksi asennettuna. Drupal vaatii käyttäjältä ja ylläpitäjältä kuitenkin eniten aikaa ja osaamista sivuston hallitsemiseen. Drupalin on sanottu olevan monipuolisin avoimen lähdekoodin sisällönhallintaohjelma, jos sitä osaa hallinta kokonaisvaltaisesti. (iMajestic 2009, Stiffler-Dean Tim 2009.)

Lyhyesti voisi vielä alleviivata kaikkien näiden kolmen sisällönhallintajärjestelmän suoranaiset käyttötarkoitukset aloittaen WordPressistä. WordPress on alun perin suunniteltu blogialustaksi ja soveltuu parhaiten pienten sivustojen ylläpitoon. Sivustoilla voi olla sekä dynaamisia, uutisten omaisia, vaihtuvia artikkeleita ja staattisia sivuja. Kuitenkin lisäosien käytöllä ja kohtalaisella XHTML/CSS- ja PHP-osaamisella sen avulla voi tehdä suuria ja monipuolisiakin sivustoja.

Joomla on suoraan käyttövalmis WWW-sisällönhallintajärjestelmä, joka on näistä kolmesta selvästi raskain sovellus, mutta johon löytyy valmiita sivupohjia ja moduuleita (vastaavat kuin WordPressin lisäosat) todella kattavasti. Joomlaan järjestelmällä käyttäjä voi tehdä näyttävät, monipuoliset sivustot, vaikka ei osaisi koodata lainkaan. Joomlaan raskaudesta kertoo muun muassa se, että

oletusasennuksessa sen tietokanta paisuu yli 50 taulun kokonaisuudeksi. Vastaavasti WordPressin ja Drupalin oletusasennuksessa tietokannan koko jää viidesosaan Joomla'n tietokannasta. Tämä ei ole tietenkään ole suoraan verrannollinen järjestelmän nopeuteen, mutta se antaa suunnan järjestelmän rakenteesta ja koodin selkeydestä. Joomla on näistä kolmesta nuorin järjestelmä, sillä sen ensimmäinen versio on julkaistu vuonna 2005. (iMajestic 2009.)

Drupal on myös suoraan käyttövalmis sisällönhallintajärjestelmä, ja se on ollut tässä vertailussa mukana olleista järjestelmistä markkinoilla pisimpään. Sen aktiivinen kehitystyö on aloitettu jo vuonna 2001. Drupal on Joomlaa nopeampi, koodiltaan siistimpi ja sillä on mahdollista tehdä monipuolisempia sivustoja kuin Joomlailla, puhumattakaan WordPressistä. Syy miksi Drupal ei ole päässyt valta-asemaan, on sen oppimiskäyrässä. Drupal on selvästi vaikein järjestelmä opetella hallitsemaan, koska siinä on todella paljon säädettävissä olevia osia alueita. Drupal onkin web-kehittäjien mielestä paras avoimen lähdekoodin työkalu WWW-sisällönhallintaan, mutta vähemmän tietotaitoa koodaamisesta omaavat käyttäjät karttavat sitä. (iMajestic 2009, Stiffler-Dean Tim 2009.)

4.3 WordPressin hallinta ja käyttöliittymä

Onnistuneen WWW-sisällönhallintaohjelman on oltava looginen ja helposti omaksuttava. Jos sivuston ylläpitäjällä on tunne, että hän ei hallitse kaikkia ohjelman elementtejä ja toimintoja, ovat ohjelman tekijät tehneet jotain väärin. WordPressinkin kohdalla tällainen pelko tulee pintaan alussa, kuten mielestäni kaikkien WWW-sisällönhallintaohjelmien kanssa, mutta WordPressin kohdalla pelon tunne muuttuu nopeasti hallinnan tunteeksi. Hallinnan tunne muodostuu siitä, että ohjelma on niin ylläpidollisesti kuin koodinkin puolesta selkeää ja loogista.

WordPress on tunnettu helppokäyttöisyydestään ja tähän on pyritty jo ohjelman käyttöönoton yhteydessä. Kuuluissa viiden minuutin asennus ei ole tuulesta temmattu ja käytännössä ohjelman asennus onkin todella helppoa. Kun WordPress tiedostot ovat ladattu ja siirretty palvelimelle, luodaan tietokanta (ellei

ole jo luotu valmiiksi webhotellipalvelun tilauksen yhteydessä) ja kirjoitetaan selaimen osoitekenttään osoite, jonne WordPress tiedostot on siirretty. Selaimeen ilmestyy ohjattu asennus, jossa tulee syöttää tietokantaa koskevat tiedot (kuvio 1).



Kirjoita alle tietokannan tiedot. Jos et ole varma yksityiskohdista, ota yhteys palveluntarjoajaasi.

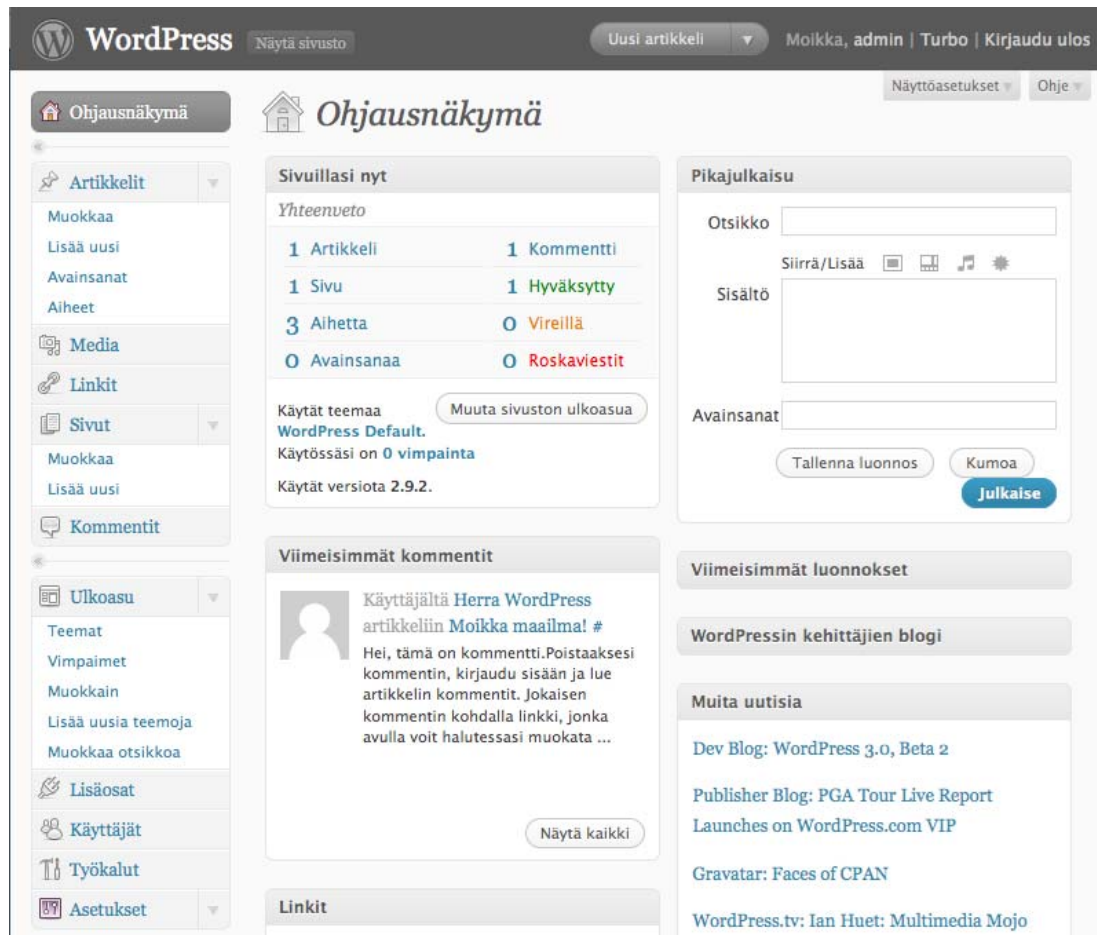
Tietokannan nimi	<input type="text" value="wordpress"/>	Sen tietokannan nimi, jota haluat käyttää WordPressin kanssa.
Tunnus	<input type="text" value="username"/>	MySQL-käyttäjätunnukseksi
Salasana	<input type="text" value="password"/>	...ja MySQL-salasanasasi.
Tietokantapalvelin	<input type="text" value="localhost"/>	99% todennäköisyydellä tätä arvoa ei tarvitse muuttaa.
Tietokantataulujen etuliite	<input type="text" value="wp_"/>	Jos haluat tehdä useamman WordPress-asennuksen samaan tietokantaan, muuta tämä joksikin muuksi.

Kuvio 1. Kuvankaappaus WordPressin asennuksen vaiheesta, jossa syötetään tietokantaa koskevat tiedot.

Tiedon syöttämisen jälkeen WordPress asentuu palvelimelle ja tarjoaa lopuksi valmiiksi luodan salasanan, jonka jälkeen järjestelmään voidaan kirjautua. Ihan ensimmäiseksi on syytä vaihtaa salasana, jonka jälkeen ollaan valmiita pyörittämään järjestelmää.

Kun ensimmäisen kerran kirjaudutaan järjestelmään, avautuu käyttäjälle WordPressin ohjausnäky (engl. Dashboard). Ohjausnäkyästä näkee sivuston yhteenvedon, käytössä olevan teeman ja käytössä olevien vimpaimien määrän sekä viimeisimmät kommentit ja luonnokset. Tarvittaessa ohjausnäkyästä voi suoraan julkaista uuden artikkelin, mutta tämä on ehkä hieman turha ominaisuus. WordPressin ylläpitoliittymässä kaikki tarvittava on

näkyvillä ylläpitosivun vasemmassa laidassa ja toimintojen otsikointi on hyvin selkeä (kuvio 2).



Kuvio 2. Kuvankaappaus WordPressin ohjausnäköymästä, kun järjestelmään kirjaudutaan ensimmäistä kertaa.

Asetuksia ei tarvitse asennuksen jälkeen juurikaan muuttaa. Tärkeimmät perusylläpitotehtävät asennuksen jälkeen ovat aikavyöhykkeen vaihtaminen, osoiterakenteen vaihtaminen sisältöä kuvailevammaksi ja etusivun ja artikkelisivun määrittäminen. Oletusarvoisesti etusivulla näytetään kymmenen uusinta artikkelia. Tällaisesta blogimaisesta rakenteesta pääsee eroon lisäämällä uuden sivun, nimeämällä sen esimerkiksi etusivuksi ja valitsemalla se näytettäväksi etusivulla, kun käyttäjä saapuu sivustolle.

Jokaisen WordPress-sivuston yhteydessä tulisi myös parantaa sen tietoturvallisuutta ja lisätä varmuuskopiointiin sekä hakukoneoptimointiin soveltuvat lisäosat. Nämä ovat yksinkertaisia tehtäviä, joihin löytyy helposti ohjeet, mutta ne parantavat sivuston toiminnallisuutta ratkaisevasti.

5 WORDPRESS – SIVUSTON SUUNNITTELU JA TOTEUTUS

Tässä luvussa käydään läpi WordPressin rakennetta, sitä miten sen teemat muodostuvat ja mitä on otettava huomioon kun suunnitellaan sivusto WordPressille. Lisäksi syvennyttään siihen, kuinka WordPressiä voi käyttää tehokkaasti WWW-sisällönhallintajärjestelmänä.

5.1 WordPressin sivurakenne

WordPress käyttää sivujen luomiseen teematiedostoja, joiden koodissa käytetään WordPressin omia teema-tagejä, eli PHP-funktioita, joilla haetaan sisältöä sivuston tietokannasta. Esimerkki teema-tagista on `<?php the_title(); ?>`, joka tuo sivun tai artikkelin otsikon siihen kohtaan, jossa tagia on käytetty sivupohjassa. Yleensä tätä tagia käytetään sivupohjassa ennen sisältöä hakevaa teema-tagia. Lisätietoa teema-tageista ja muista WordPressin sivupohjien funktioista on tiivistetysti liitteessä 5.

Teknisesti WordPress ei tarvitse kuin kaksi teematiedostoa toimiakseen, jotka ovat `index.php` – ja `style.css` –tiedostot. Yleinen rakenne sivulle sisältää kuitenkin vähintään 4 teematiedostoa, jotka ovat `index.php`, `header.php`, `sidebar.php` ja `footer.php`. Tällainen rakenne on esimerkiksi WordPressin mukana tulevan vakioteeman etusivulla.

Sivun ydinrakenne heijastaa WWW-sivun runkoelementtejä, eli `div`-tageja, joiden sisälle sivun sisältö tuodaan. Ydinrakenne muodostuu minimissään ylä- ja alatunnisteesta (engl. header and footer), sisältöelementistä ja navigaatioelementistä tai sivupalkista. Nämä ovat loogisia, kun rakennetta vertaa WordPressin vakioteemassa olevan sivun runkoelementteihin. Vakioteeman sivurunko on seuraavanlainen:

```
<body>
<div id="page">
    <div id="header"></div>
    <div id="content" class="narrowcolumn"></div>
    <div id="sidebar"></div>
```

```

        <div id="footer"></div>
    </div>
</body>

```

Tässä esimerkkikoodissa sivu on yhden runkoelementin, eli page-divin sisällä. Tämä page-divi sitoo sisäänsä header-divin, eli ylätunnisteen; content-divin, eli sisältöelementin; sidebar-divin, eli sivupalkin ja footer-divin, eli alatunnisteen.

5.1.1 WordPressin teemahierarkia

WordPress käyttää sivun luomisessa tietynlaista teemahierarkiaa, jonka perusteella se määrittelee, mikä teematiedosto on ensisijainen.



Kuvio 3. Pelkistetty kuva WordPressin käyttämästä teemahierarkiasta. (mukaillen WordPress Codex 2010c)

Kuviosta 3 käy ilmi WordPressin teemahierarkkia päätasolla. WordPress toimii siten että esimerkiksi etusivulle tultaessa, tarkistetaan tiedoston home.php

olemassaolo, jos kyseistä tiedostoa ei ole, ladataan index.php. Samoin esimerkiksi sisältösivulle tultaessa tarkistetaan, onko sivulle määritetty omaa teemaa (pagename.php), jos sivulle ei löydy omaa teemaa, ladataan page.php tiedosto, joka määrittää pohjan kaikille staattisille sivuille. Jos kumpaakaan ei löydy, haetaan jälleen index.php. Samanlainen silmukka käydään läpi muidenkin sivujen osalta.

5.1.2 Sivupohjan rakenteen suunnittelu

Sivupohjan rakennetta suunniteltaessa on mietittävä, tarvitaanko esimerkiksi hakusivua tai yksittäisen artikkelin esittämistä. Jos tehdään sivusto, joka sisältää pelkästään staattisia sivuja, voidaan artikkeleiden näyttämiseen tarkoitetut teematiedostot jättää rauhaan. Tällöin säästetään aikaa, kun ei tarvitse alkaa suunnittelemaan tarpeettomien sivujen esitystapaa. Jos myöhemmin päätetään lisätä esimerkiksi dynaaminen uutissivu sekä sivuston sisältöhaku, ne voidaan suunnitella ja ottaa käyttöön vasta kun niille on tarvetta.

Teemahierarkkia siis mahdollistaa ulkoasun rakentamisen vaiheittain ja lisäksi monipuolisesti ja yksilöidysti. Huolellisesti rakennetussa teemassa on esimerkiksi otettu huomioon, miltä mahdollisesti 404-virhesivu näyttää. 404-sivu tulee esiin, kun sivustolla on väärään paikkaan osoittava linkki tai sivua ei enää ole olemassa. 404-sivua ei ihannetapauksessa tule koskaan esille, mutta jossain vaiheessa se todennäköisesti tulee ja tällöin on hyvä näyttää sivu, jossa pahoitellaan virhettä ja kerrotaan, mikä meni mahdollisesti pieleen. 404-sivun merkitys voi olla pieni, mutta se antaa ammattimaisen kuvan sivustosta. Se kertoo, että sivuston käyttäjät on otettu huomioon ja heitä halutaan palvella mahdollisimman hyvin.

Riippuen projektista, lähes aina huomioon otettavia teematiedostoja ovat sivuston sisältöhaun tulossivu, yksittäisen artikkelin ja sivun näkymä sekä se, millälailla artikkeleiden kategoria ja arkistolistaus näytävät.

5.2 Sivupohjan rakentaminen ja ulkoasun muokkaaminen

Sivupohjan rakentaminen edellyttää tietynlaisia toimia ennen varsinaista toteutusta. On syytä kartoittaa mahdollisimman tarkasti, mitä sivustolta halutaan. Millaisia elementtejä se voisi sisältää ja millainen on sivuston sisällön rakenne. Sivusto voi sisältää pelkästään staattisia sivuja, se voi olla pelkistetty blogi tai se voi olla näiden yhdistelmä. Kun tiedetään, millaista sisältöä sivustolle on tulossa ja tiedetään suurinpiirtein, miten sisältö aiotaan esittää, voidaan miettiä sivupohjan rakennetta.

Itse olen huomannut, että suunnittelu voi kulkea toisinkin päin. Kiireisessä mainostoimistossa suunnittelu etenee toisinaan siten, että aluksi kartoitetaan asiakkaan kanssa sivuston käyttötarkoitusta ja sisältöä. Kartoituksen jälkeen työ siirtyy ulkoasun suunnittelijoille, jotka miettivät sivuston ulkoasun lisäksi sivuston rakennetta ja vasta tämän jälkeen sivusto tulee koodattavaksi. Tällä tavoin toteutettuna sivuston loogisuuden ja rakenteen kanssa voi tulla ongelmia toteutusvaiheessa.

Koska WordPress tuottaa validia XHTML/CSS –koodia, voidaan sille rakennettu sivusto myös suunnitella tavallisen XHTML/CSS –sivuston tavoin. Itse asiassa voisi ajatella, että WordPress toimii käyttöliittymänä ja tietopankkina, josta tietoa jaetaan julkiselle puolelle (frontend) teema-tagien avulla. Teeman tarkoitus on antaa tiedolle raamit.

Esimerkkinä WordPressin vakioteeman sivun näyttämisen (page.php) koodi:

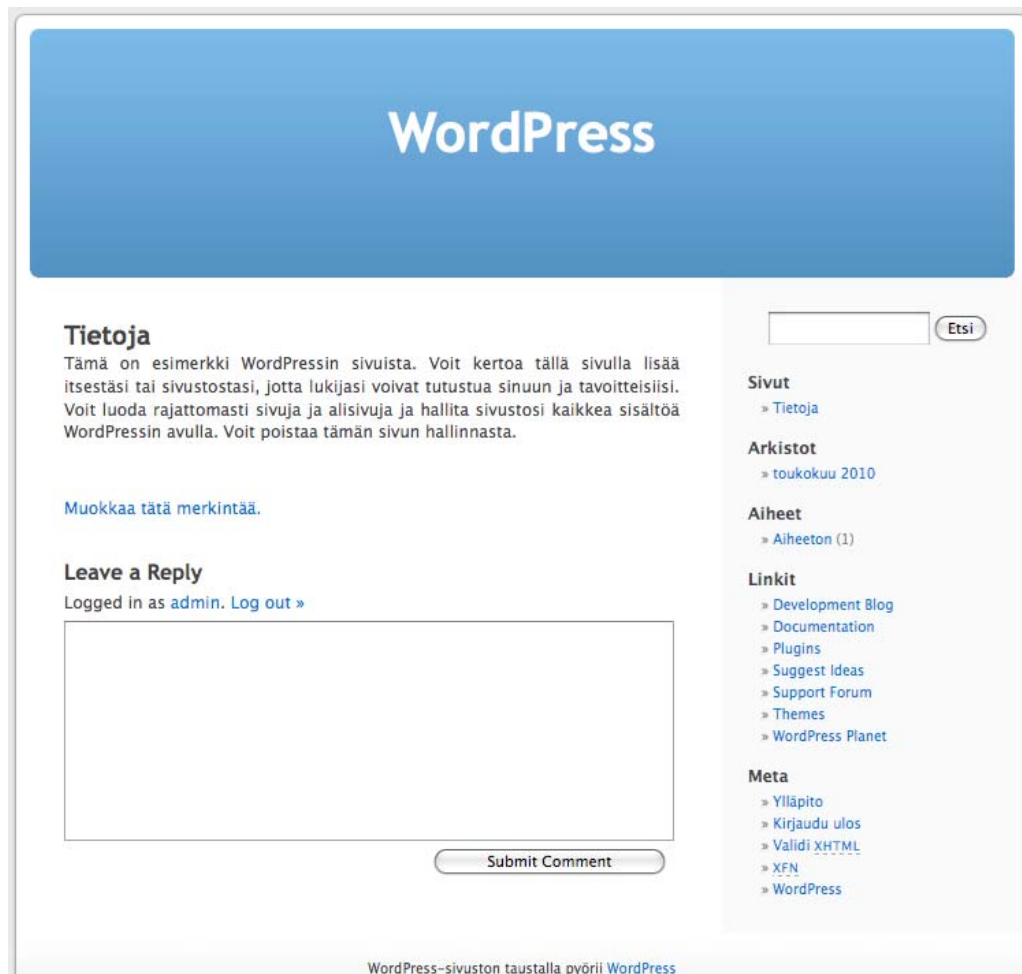
```
<?php get_header(); ?>
<div id="content" class="narrowcolumn" role="main">
  <?php if (have_posts()) : while (have_posts()) : the_post(); ?>
    <div class="post" id="post-<?php the_ID(); ?>">
      <h2><?php the_title(); ?></h2>
      <div class="entry">
        <?php the_content('<p class="serif">' . __('Read the rest
of this page &raquo;', 'kubrick') . '</p>'); ?>
```

```

        <?php wp_link_pages(array('before' => '<p><strong>' .
        __('Pages:', 'kubrick') . '</strong> ', 'after' => '</p>',
        'next_or_number' => 'number')); ?>
    </div>
</div>
    <?php endwhile; endif; ?>
    <?php edit_post_link(__('Edit this entry.', 'kubrick'), '<p>', '</p>'); ?>
    <?php comments_template(); ?>
</div>
    <?php get_sidebar(); ?>
    <?php get_footer(); ?>

```

Koodi on otettu tähän malliksi, jotta voidaan esittää esimerkin avulla sivun muodostuminen. Punaisella värjätty koodi on WordPressin toiminnallisuuteen vaikuttavaa koodia, ne ovat PHP:tä, ja tässä tapauksessa WordPressin teematageja, joilla tehdään hakuja tietokantaan ja muihin teematiedostoihin. Mustalla värjätty koodi puolestaan on XHTML-koodia, jonka on tarkoitus luoda sivulle raamit ja antaa sille yhdessä CSS-tyylitiedoston avulla ulkoasu.



Kuvio 4. Kuvankaappaus WordPressin Tietoja-sivusta ensiasennuksen jälkeen.

Kuviossa 4 on WordPressin Tietoja-sivun näkymä oletusasennuksen jälkeen. Käytössä on siis WordPressin oletusteema "Kubrick". Kubrick-teema on hyvin selkeä ja miellyttävän seesteinen ulkoasullisesti. Tarkoitus on kuitenkin selvittää, mitä tapahtuu kun XHTML-raamit poistetaan sivulta. Kuvio 5 on kuvankaappaus samasta sivusta, joka on kuviossa 4, sillä erolla että XHTML-elementit on poistettu sivun teematiedostoista. Selkeyden vuoksi ``- ja ``-tagit, sekä otsikkotagit on jätetty paikoilleen, mutta muuten kaikki sivun runkoelementit on poistettu ja tiedot haetaan sivuille peräkkäin siinä järjestyksessä kuin WordPressin teematagit ovat sivun teematiedostoissa määritelty. Kuvioita 4 ja 5 vertaamalla nähdään, että kaikki oleellinen tieto löytyy

molemmista sivuista, mutta kuvion 4 kuvankaappauksessa tieto on asetettu raameihin, jotka on sijoitettu loogisesti. Ajatusta voisi verrata aikakausilehden taittoon. Jos aikakausilehdessä olisi 100 sivun verran tekstiä ilman minkäänlaista tekstin sijoittelua ja muotoilua, olisi se todella pitkästyttävä lehti lukea, vaikka sisältö olisikin mielenkiintoista.

[WordPress](#)

Tietoja Tämä on esimerkki WordPressin sivuista. Voit kertoa tällä sivulla lisää itsestäsi tai sivustostasi, jotta lukijasi voivat tutustua sinuun ja tavoitteisiisi. Voit luoda rajattomasti sivuja ja alisivuja ja hallita sivustosi kaikkea sisältöä WordPressin avulla. Voit poistaa tämän sivun hallinnasta. [Muokkaa tätä merkintää.](#)

Leave a Reply

Logged in as [admin](#). [Log out »](#)

• Etsi

• Sivut

◦ [Tietoja](#)

• Arkistot

◦ [toukokuu 2010](#)

• Aiheet

◦ [Aiheeton](#) (1)

Kuvio 5. Kuvankaappaus WordPressin Tietoja-sivusta runkoelementtien poistamisen jälkeen.

WordPressin sivupohjan rakentamisen työvaiheet sisältävät järjestyksessään seuraavat vaiheet: sivuston sisällön kartoitus, sivuston rakenteen suunnittelu, ulkoasun suunnittelu ja ulkoasun taittaminen XHTML/CSS-sivuiksi. Näiden vaiheiden jälkeen voidaan alkaa istuttamaan sivustoa WordPress sisällönhallintaohjelmaan, joka tapahtuu teematiedostojen ja niiden sisältämien teematagien avulla.

5.3 WordPress-lisäosat

WordPress on monipuolinen ja tehokas blogialusta käytettäväksi sellaisenaan, mutta sen käyttö WWW-sisällönhallintaohjelmaksi on hieman vajavaista, ellei lisäosia oteta käyttöön. Lisäosien esittely voi näyttää hieman irralliselta muuhun työn sisältöön verrattuna, mutta se on mielestäni oleellinen osa WordPressin käytettävyyttä ja niiden käyttö mahdollistaa paljon sisällönhallintaan liittyviä ratkaisuja, joihin WordPress ei oletusarvoisesti kykene.

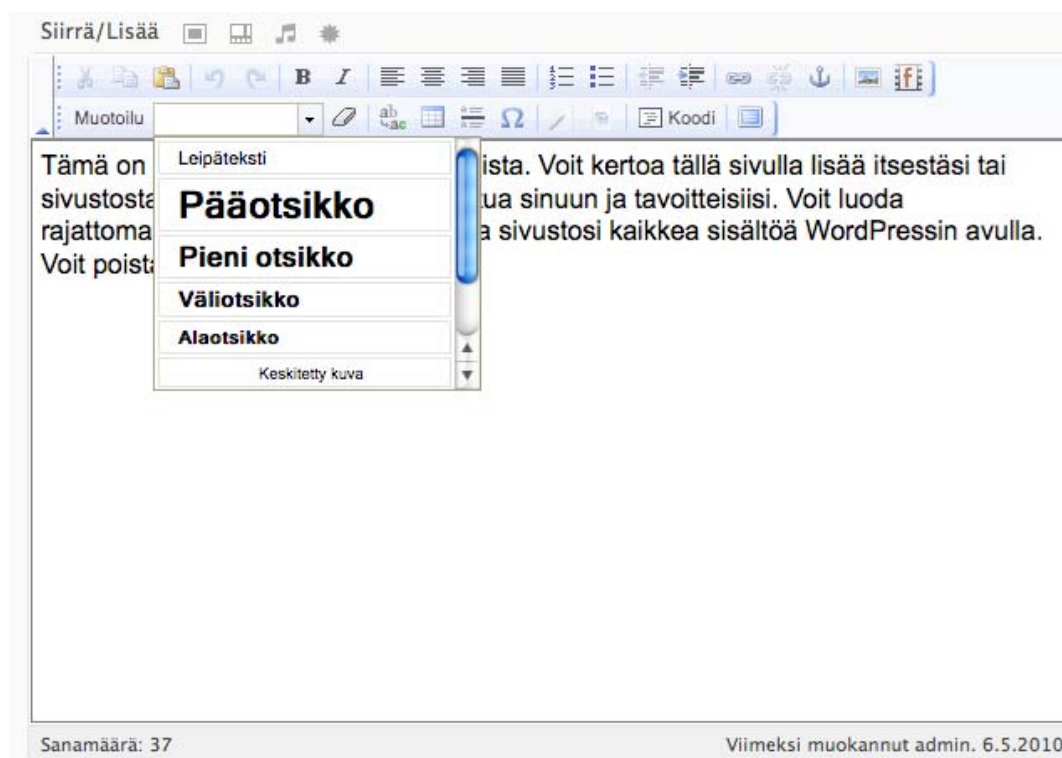
Tässä työssä esittelen omakohtaisen kokemuksen sekä tutkimustyön pohjalta koottuja lisäosia, jotka parhaiten parantavat WWW-sivuston sisällönhallintaa. Lista ei missään nimessä ole absoluuttisesti oikea ja ainoa, koska uusia ja mahdollisesti parempia lisäosia kehitetään jatkuvasti lisää. Lisäksi lista on koottu paljolti omien mieltymysten ja käyttäjäkokemusten perusteella.

Lisäosia moititaan usein siitä, että ne ovat huolimattomasti koodattuja PHP-viritelmiä, jotka jättävät jälkeensä paljon tietoturva-aukkoja ja tekevät sivustosta raskaan. Tämä väite pitää osittain paikkansa, sillä yli 9000 lisäosan verkostoon mahtuu todella huonostikin koodattuja lisäosia. Lisäksi kaikki lisäosat eivät ole yhteensopivia keskenään, joten tietyn lisäosan asennuksen jälkeen jokin toinen lisäosa voi lakata toimimasta tai pahimmassa tapauksessa aiheuttaa PHP-virheen ja sivuston kaatumisen. Onneksi WordPressin yhteisöverkoston lisäosien luokittelumalli (engl. rating) ja tukifoorumi auttavat parhaiden ja yhteensopivien lisäosien valinnassa. Tässä työssä esiteltyt lisäosat ovat jaoteltu WWW-sivujen ylläpidon kannalta tärkeimpien osa-alueiden mukaan. (WordPress.org 2010a.)

5.3.1 Tekstin muokkaus ja lomakkeet

Foliopress WYSIWYG on vaihtoehtoinen ja parempi tekstieditori WordPressiin. Foliopress WYSIWYG antaa mahdollisuuden lisätä helposti omia tekstityylejä, jotka tulevat pudotusvalikkoon normaalin otsikkomallin jälkeen (kuvio 6). Tämä lisäosa tuo mukanaan myös videoiden lisäämisen, kun asentaa rinnalle FV Wordpress Flowplayer:in. Kuitenkin tärkeimmät ominaisuudet CMS-käyttöä ajatellen ovat omien tyylien määrittäminen, tekstin suoraan liittämisen (ctrl+v)

estäminen ja todella hyvin toimiva taulukoiden lisääminen ja editoiminen. Kaikki nämä ominaisuudet helpottavat sivuston ylläpitäjää pitämään sivuston koodin mahdollisimman puhtaana ja yleistyylin mahdollisimman samanlaisena. Koska otsikkotyylit ovat lukittu ja oletuksena tekstin väri- ja kokomuutostyökalut ovat piilotettu, ei sivuston päivittäjä pääse tekemään mitään niin sanotusti ylimääräistä. Lisäksi Foliopress WYSIWYG:n työkaluvalikkoa ja teemaa on helppo muuttaa, jos tälle on projektikohtaisia tarpeita. (Foliovision 2010.)



Kuvio 6. Kuvankaappaus sivun editoinnista WordPressissä, jossa on käytössä Foliopress WYSIWYG –tekstieditori valmiiksi konfiguroiduilla tekstityyleillä.

Cforms II ja Contact form7 lisäosat auttavat luomaan helposti monipuolisia lomakkeita, joita voi käyttää sivuilla. Näistä kahdesta cformsII on monipuolisempi, mutta ei niin helppokäyttöinen kuin Contact form 7. Molemmilla voi tehdä helposti yksinkertaisia palautelomakkeita tai kyselylomakkeita ja molempien ulkoasua voi suhteellisen helposti muokata. CformsII on kuitenkin selkeästi tehty laajempienkin lomakkeiden suunnitteluun ja siitä löytyy sisäänrakennettuna muun muassa monisivuisten lomakkeiden teko ja lomaketeemat, joita voi

monipuolisesti muokata CSS-tyyleillä. Esittelen tässä molemmat lisäosat johtuen niiden erilaisesta käyttötarkoituksesta. Contact form 7 on niin helppokäyttöinen, että sillä voivat asiakkaatkin tehdä omia lomakkeitaan, kun siihen annetaan oikeudet. CformsII on puolestaan todella hyvä työkalu ylläpitäjille monipuolisempien lomakkeiden luomiseen.

5.3.2 Sisällönhallintalisäosat

Capability manager on yksi monista käyttäjäroolien hallintaan tehdyistä lisäosista. Tällä lisäosalla voi hallita kaikkien WordPressin vakioasetuksessa tulleita käyttäjärooleja ja vaihtoehtoisesti lisätä myös uusia käyttäjärooleja. Lisäosassa on myös käyttäjäroolien varmuuskopiointi ja mahdollisuus luoda palautuspiste, jos jokin menee pieleen. Kun tehdään sivustoja, jonka ylläpitoon annetaan usealle käyttäjälle eri tason oikeuksia, on tämä lisäosa todella tärkeä osa sisällönhallintaa.

WPML Multilingual CMS on kieliversioiden hallintaan tehty lisäosa, joka on todella monipuolinen ja tehokas. Tällä lisäosalla voi tehdä sivuston kieliversiot kokonaisvaltaisesti, ei pelkästään tekstin ja valikoiden osalta, vaan myös muiden elementtien, kuten kuvien ja videoiden osalta. Kaikki tapahtuu WordPressin ylläpitopuolelta, jossa voidaan skannata teemasta käännettävät merkkijonot. Huomioitavaa on, että sivuston teema on rakennettava tukemaan kieliversioita. WPML Multilingual CMS – lisäosa käyttää teemojen ja sisällön lokalisointiin WordPressin `__()`- ja `_e()`-funktioita, jotka pohjautuvat GNU gettext lokalisaatio-alustaan (engl. framework). Tämä kyseinen alusta on muodostunut standardiksi avoimen lähdekoodin ohjelmien keskuudessa ja se on käytössä hyvin laajasti. Nämä funktiot toimivat siten, että kun jokin teksti halutaan valmistella myöhemmin käännettäväksi, kirjoitetaan se funktion sisään. (GNU gettext 2010, WordPress Codex 2010d.)

Funktioiden ero on se, että `__()` hakee tekstit PHP:n `return`-lauseella ja sitä käytetään kun palautetaan muuttuja toiseen funktioon, `_e()` funktiolla puolestaan tulostetaan `echo`-lauseella suoraan sivulle. Esimerkiksi WordPressin

vakioteemasta löytyy sivun PHP-koodista `<p><?php _e('Sorry, no posts matched your criteria.');`; `?></p>`. Tämä kertoo sen, että kappalemallin sisälle tulostetaan echo-lauseella "Sorry, no posts matched your criteria", joka on mahdollista lokalisoida eri kielille. Toinen esimerkki on myös vakioteemassa sijaitsevan sivun PHP-koodista, jossa artikkeleiden navigaatiofunktion muuttujista tehdään lokalisoitavia `__()`-funktion avulla: `<?php posts_nav_link(' — ', __('« Newer Posts'), __('Older Posts »'));`; `?>` (WordPress Codex 2010d).

Tämä lisäosa pystyy käyttämään myös suoraan WordPress-yhteisön tekemiä kieliversioita, jotka kääntävät WordPressin ylläpitopuolen ja vakioteeman. Näitä yhteisön tekemiä ammattimaisia kieliversioita on saatavilla tällä hetkellä 67, jotka kattavat muun muassa Kiinan ja Hindin. (WordPress Codex 2010e).

Display widgets mahdollistaa widgettien, eli suomeksi hauskesti käännettyjen "vimpaimien" paremman hallinnan. Tämä lisäosa mahdollistaa ylläpidon puolelta vimpaimien esittämisen vain tietyillä sivuilla. Vimpaimen yhteyteen tulee ylläpitopuolella valintaruudukko, josta voi valita millä sivulla tai sivuilla vimpain on näkyvillä. Lisäosa tulee tarpeen, kun halutaan tehdä yksinkertainen muutos tietyn sivun ulkoasuun, mutta sivulle ei kuitenkaan haluta tehdä omaa sivupohjaa.

Exclude Pages from Navigation – lisäosa on tärkeä CMS-käytössä, sillä sen avulla on mahdollista piilottaa sivuja navigointivalikosta. Lisäosan asennuksen jälkeen sivujen muokkauksen yhteyteen tulee valintaruutu, josta voi valita näkyykö sivu navigointivalikossa vai ei. WordPressin käytön rajat sisällönhallintajärjestelmänä tulevat helposti esiin juuri navigoinnin hallinnan yhteydessä, sillä ilman lisäosia sivujen piilottaminen navigointivalikosta onnistuu vain teematagin konfiguroinnilla.

5.3.3 Hakukoneoptimointi

Kun markkinointi ja näkyvyys ovat siirtyneet yhä enemmissä määrin Internetiin, ei riitä että sivut ovat olemassa, jos niitä ei kukaan tai varsinkaan hakukoneet löydä. Hakukoneoptimointi on todella tärkeä osa-alue, johon kannattaa panostaa. Yksi

hyvistä puolista WordPressissä on sen suhteellisen selkeä ja hakukoneystävällinen osoiterakenne, mutta tullakseen näkyväksi WWW-sivujen viidakossa, pitää sitä hieman parantaa.

All in One SEO Pack on yksi parhaista lisäosista hakukoneoptimointiin. Se parantaa automaattisesti sivuston hakukonenäkyvyyttä muokkaamalla sivujen otsikot hakukoneystävällisempään muotoon ja lisäämällä automaattisesti sivuille kuvauksen (engl. Description) ja avainsanat. Otsikoita, kuvausta ja avainsanoja voi myös muokata sivukohtaisesti. Lisäksi All in One SEO Pack tarkistaa sivustolta sisältökopiot ja piilottaa ne. Sisällön kopioinnista hakukoneet rankaisevat huonommilla sijoituksilla tai poistamalla ne kokonaan listauksista.

Redirection lisäosa on hyödyllinen, kun jokin linkki menee sivustolla rikki, eli se ohjaa sivulle jota ei ole tai sivulle joka on siirretty eri paikkaan. Toisinaan sivujen kestolinkin nimeä tai sivuston rakennetta muutetaan. Esimerkiksi voi olla sivu, joka sijaitsee osoitteessa www.sivusto.fi/esimerkki ja se muutetaan olemaan uuden sivun alisivu www.sivusto.fi/esimerkit/esimerkki. Tällöin linkki jossain ulkoisella sivulla tai jopa sisälinkki ohjaa väärään paikkaan. Redirection osaa seurata näitä muutoksia ja ohjaa liikenteen automaattisesti osoitteesta www.sivusto.fi/esimerkki osoitteeseen www.sivusto.fi/esimerkit/esimerkki.

6 MALLISIVUSTON TOTEUTUS

6.1 Lähtökohdat

Tämän opinnäytetyön yhteyteen kuuluu demo-, eli mallisivusto, joka tehtiin luonnollisesti WordPress-sisällönhallintaohjelmaa käyttäen. Sivuston on tarkoitus toimia yhtenä alustavana sivupohjana uusien sivustojen luomisvaiheessa. Sivustolle on luotu mahdollisimman paljon erilaisia dynaamisia ja staattisia elementtejä, joiden avulla on pyritty tekemään mahdollisimman monipuolinen, elävä ja toimiva sivusto.

Sivuston teema on rakennettu selkeäksi toiminnallisuudeltaan, rakenteesta on luotu erillinen dokumentti ja siinä käytettyjen teematiedostojen koodi on kommentoitu hyvin. Selkeydellä, kommentoinnilla ja dokumentoinnilla on pyritty saamaan mahdollisimman hyvä runko uusia WordPress-sivustoja silmällä pitäen.

Mallisivustossa on pyritty ottamaan huomioon WordPress-sivuston hallinta kokonaisvaltaisesti, joten ylläpidon puolelle on asennettu mahdollisimman hyviä sisällönhallintaan, hakukoneoptimointiin ja turvallisuuteen sekä varmuuskopiointiin liittyviä lisäosia. Lisäksi sivustolle on tehty valmiita koodipätkiä (engl. code snippets), joiden käyttötarkoitus ja -ohje on dokumentoitu.

6.2 Sivuston toteutus

Sivusto toteutettiin 960 grid system CSS-alustaa pohjana käyttäen. 960 grid system on Nathan Smithin kehittämä CSS-alusta, joka on nimensä mukaan 960 pikseliä leveä HTML-sivupohja, jossa on luotu CSS-tyyleillä 12 palstainen rakenne. Pohjaan sisältyy CSS-tyylin nollaus, jonka tarkoitus on asettaa esimerkiksi rivivälit, marginaalit, kuvien reunukset ja fonttien määritykset oletusarvoisesti kaikille selaimille samanlaisiksi. CSS-alustan tarkoitus on nopeuttaa sivupohjan rakentamista, koska siitä pystyy helposti muokkaamaan kahden tai kolmen palstan sivupohjia ja sen toimivuus on valmiiksi testattu yleisimmillä selaimilla. (960 grid system 2010, Sonspring 2008.)

Sivusto on rakennettu viimeisimmän julkistetun version päälle, joka on versio 2.9.2. Sivustolla on käytössä kaikki, jo aikaisemmin tässä työssä esitelty lisäosat ja sivustolla on valmiiksi määritettynä usean eri tason käyttöoikeuksia. Sivustoa rakennettaessa on otettu huomioon hakutuloksien esittäminen, 404-virhesivu, artikkeleiden arkistointi, sosiaalinen media facebookin ja twitterin osalta, kieliversiointi, javascript-toiminnallisuus sekä erilaisten medioiden, kuten kuvagallerioiden ja videoiden esittäminen.

Sivupohjien koodi on kommentoitu mahdollisimman hyvin sivupohjien jatkokäyttöä ajatellen. Lisäksi sivustolla käytetyt tekniikat sekä lisäosien mahdolliset muokkaamiset ja muut sivuston toimintaan liittyvät asiat on dokumentoitu, jotta niiden käyttöönotto muilla sivustoilla olisi joustavampaa.

6.3 Käyttötarkoitus

Mallisivuston toivotaan selkeyttävän Mainostoimisto Seven-1:n työnkulkua WWW-projektien läpiviemisessä. Kun parhaat mahdolliset lisäosat ja selkeä, helposti muokattavissa oleva runko sivupohjalle on luotu, ei sitä tarvitse tehdä joka kerta uudelleen. Työnkulun selkeytymisestä olisi hyötyä ajallisesti ja laadullisesti toteutusvaiheessa, kun voidaan käyttää aiemmin hyväksi havaittuja menetelmiä hyödyksi. Työnkulun selkeyttäminen auttaa myös suunnittelu- ja tarjousvaiheessa, kun voidaan arvioida entistä tarkemmin työhön kuluva aika.

Tämän opinnäytetyön yhteydessä luotu mallisivusto tulee toivottavasti toimimaan lähtökohtana monessa uudessa Seven-1:n WWW-projektissa. Jos kaikki menee hyvin, on yhtiöllä tarkoitus jatkaa mallisivuston kehittämistä soveltumaan yhä vaativimpiin ja omalaatuisimpiin WWW-projekteihin.

7 YHTEENVETO

Opinnäytetyö oli opettava prosessi, joka omalta kohdaltani eteni todella nopeassa syklissä. Toimeksianto työlle tuli helmikuun lopussa ja kunnolla työ pääsi pyörähtämään käyntiin vasta reilusti maaliskuun puolella. Lisäksi tein opinnäytetyötäni päivätyön ohessa ja työnkuva muuttui moneen otteeseen työn aloittamisen jälkeen. Yleinen mielikuva tästä prosessista oli kuitenkin positiivinen. Työ oli aiheeltaan mielenkiintoinen ja sopivan haastava. Lisäksi työn havainnollinen tulos, eli WordPress-alustan päälle rakennettu mallisivusto dokumentteineen oli onnistunut.

Aihetta käsittelevää kirjallisuutta oli mielestäni vaikea löytää ja varsinkin kirjastojen kokoelmissa sen määrä oli lähes olematonta. Yleensä löydetty kirjallisuus sivusi aihetta tai käsitteli sitä liian yleisesti. Sopivaa kirjallisuutta olisi ollut enemmän saatavilla englanniksi, mutta lähinnä tilaamalla sitä yhdysvalloista, jossa aihetta käsittelevää kirjallisuutta on parhaiten. Omalta osaltaan tiedonhankintaa vaikeutti tietysti myös kiireinen aikataulu, joka pakotti käyttämään lähes yksinomaan verkkolähteitä. Verkkolähteiden suurimmaksi ongelmaksi muodostui tiedon luotettavuus, jonka vuoksi internet-lähteiden laatu piti tarkistaa useampaa sivustoa käyttäen.

Vaikeimmaksi osa-alueeksi työn kannalta osoittautui yleinen WWW-sisällönhallintajärjestelmien taustoitus ja määrittely. Itse sivuston suunnittelu ja toteutus eivät tuottaneet ongelmia, koska olen työskennellyt WordPress-alustan parissa jo yli puoli vuotta lähes päivittäin. Myös WordPress-sivuston suunnittelun ja toteutuksen läpikäyminen oli sujuvaa ja mielenkiintoista.

Tässä työssä saavutettujen tulosten merkitys toimeksiantajalle tulee olemaan merkittävä, jos työssä käsitellyt asiat onnistutaan ottamaan sujuvasti osaksi projektien työnkulkua. Työn tulokset nopeuttavat uusien projektien läpiviemistä ja auttavat määrittämään ne projektit, joihin WordPress soveltuu parhaiten. Mielenkiintoista nähdä, kuinka hyvin tämän työn tulokset otetaan käyttöön toimeksiantajan yhtiössä ja kuinka työn tuloksia lähdetään jatkokehittämään.

Oma roolini työn tulosten käyttöönotossa ja jatkokehityksessä on varmasti merkittävä, joten senkin kannalta on hyvä, että prosessi oli loppujen lopuksi onnistunut.

LÄHTEET

- 2K Mediat 2010. Miksi PHP? Osoitteessa
www.2kmediat.com/php/johdanto2.asp. 6.5.2010.
- 960 grid system 2010. Osoitteessa www.960.gs/. 24.5.2010.
- Boiko, Bob 2005. Content Management Bible, 2nd Edition. Indiana: Wiley Publishing, Inc.
- CMS Calender 2010. History of CMS. Osoitteessa
www.cmscalendar.com/glossary.html?term=HistoryOfCMS.
 16.4.2010.
- CMS Wire 2009. Report: The Most Popular Open Source CMS, and Then Some. Osoitteessa www.cmswire.com/cms/web-cms/report-the-most-popular-open-source-cms-and-then-some-005789.php. 23.5.2010.
- Foliovision 2010. True Wordpress WYSIWYG: Foliopress. Osoitteessa foliovision.com/seo-tools/wordpress/plugins/wysiwyg. 24.5.2010.
- GNU gettext 2010. Introduction to gettext. Osoitteessa
www.gnu.org/software/gettext/gettext.html. 26.4.2010.
- iMajestic 2009. Comparing the Three Major CMS's: Drupal, Joomla and WordPress. Osoitteessa
www.imajestic.com/blog/2009/07/comparing-the-three-major-cms%E2%80%99s-drupal-joomla-and-wordpress/. 7.5.2010.
- Matt Mullenweg 2010. Sun, Oracle, WordPress, and MySQL. Osoitteessa
ma.tt/2009/04/oracle-and-open-source/. 20.4.2010.
- MySQL 2010. MySQL 5.5 Reference Manual. MySQL-ohjekirja. Osoitteessa
dev.mysql.com/doc/refman/5.5/en/index.html. 24.4.2010.
- Oracle 2010. Overview and Frequently Asked Questions. Osoitteessa
go2.wordpress.com/?id=725X1342&site=saviorrodrigues.wordpress.com&url=http%3A%2F%2Fwww.oracle.com%2Fus%2Fsun%2F038563.pdf&sref=http%3A%2F%2Fsaviorrodrigues.wordpress.com%2F2009%2F10%2F29%2Foracle-unveils-plans-for-mysql-glassfish-netbeans-and-openoffice%2F. 6.5.2010.
- PHP: Manual 2010. What can PHP do? Osoitteessa
fi.php.net/manual/en/intro-whatcando.php. 24.4.2010.

- Rob Prideaux 2004. Knowing when you need a CMS. Osoitteessa www.techsoup.org/learningcenter/webbuilding/archives/page9378.cfm. 6.5.2010.
- SEOMoz 2010. Beginner's Guide to Search Engine Optimization. Osoitteessa www.seomoz.org/article/beginners-1-page. 16.4.2010.
- Sonspring 2008. 960 grid system. Osoitteessa sonspring.com/journal/960-grid-system. 24.5.2010.
- StatCounter 2010. Global Stats. Osoitteessa gs.statcounter.com/. 23.5.2010.
- Stamatiou, Paul 2006. 5 Ways to Speed Up Your Site. Osoitteessa paulstamatiou.com/5-ways-to-speed-up-your-site
- Stiffler-Dean, Tim 2009. Wordpress vs Joomla vs Drupal. Osoitteessa www.goodwebpractices.com/other/wordpress-vs-joomla-vs-drupal.html. 23.5.2010.
- Valdrighi Michael 2003. B2 Cafelog. Osoitteessa cafelog.com. 16.4.2010.
- W3C 2010. Cascading Style Sheets. Osoitteessa www.w3.org/Style/CSS. 24.4.2010.
- W3C 2010. HTML 4.01 Specification. Osoitteessa www.w3.org/TR/REC-html40. 24.4.2010.
- Webhotellit.com 2010. Webhotellien vertailua. Osoitteessa www.webhotellit.com. 26.4.2010.
- Webhotellit24.fi 2010. Webhotellien vertailua. Osoitteessa www.webhotellit24.fi/webhotellit.html. 26.4.2010.
- Webstandards.org 2010. What we're doing. Osoitteessa www.webstandards.org/action/. 23.5.2010.
- Webmonkey.com 2008. HTML 5 Won't Be Ready Until 2022. Yes, 2022. Osoitteessa www.webmonkey.com/2008/09/html_5_won_t_be_ready_until_2022_dot_yes__2022dot/
- Wheeler, David A. 2007. Forking. Osoitteessa www.dwheeler.com/oss_fs_why.html#king. 18.4.2010.
- Wikipedia 2010. History of Internet Explorer. Osoitteessa en.wikipedia.org/wiki/History_of_Internet_Explorer. 23.5.2010

- Wikipedia 2010. List of content management systems. Osoitteessa en.wikipedia.org/wiki/List_of_content_management_systems. 15.4.2010.
- WordPress.org 2010. Extend WordPress. Osoitteessa wordpress.org/extend. 3.5.2010.
- WordPress.org 2010. Releases Category Archive. Osoitteessa wordpress.org/development/category/releases/. 3.5.2010.
- WordPress.org 2010. WordPress Blog. Osoitteessa wordpress.org/development/. 3.5.2010.
- WordPress Codex 2010. Design and Layout. Osoitteessa codex.wordpress.org/Main_Page. 4.5.2010.
- WordPress Codex 2010. Template Hierarchy. Osoitteessa codex.wordpress.org/Template_Hierarchy. 26.4.2010.
- WordPress Codex 2010. Translating WordPress. How to localize WordPress. Osoitteessa codex.wordpress.org/Translating_WordPress. 26.4.2010.
- WordPress Codex 2010. WordPress in your language. Osoitteessa codex.wordpress.org/WordPress_in_Your_Language. 26.4.2010.
- WordPress theme markets.com 2010. History of WordPress – Then and Now. <http://wordpressthemesmarket.com/History%20of%20Wordpress.html>. 16.4.2010.
- XHTML 1.0: The Extensible HyperText Markup Language (Second Edition). Osoitteessa www.w3.org/TR/xhtml1/xhtml1.pdf. 24.4.2010.

LIITTEET

Seven-1:lle luodun mallisivuston etusivu	Liite 1
Acid2- ja Acid3-testitulokset	Liite 2
Selaintilastot graafisesti esitettynä	Liite 3
Joomlan ja drupalin ylläpitopaneelin kuvankaappaukset	Liite 4
Wordpressin sivupohjien funktiot tiivistettynä	Liite 5

Rollo Action

"The excitement comes from nature"

Hae sivulta...

ETUSIVU AKTIVIITEITIT MAJOITUS TEE VARAUS YRITYS YHTEYSTIEDOT

Koskiseikkailu alk. 50€/hlö

SEIKKAILUPAKETTI

- Chilling UUTTA!
- Fespiisi!
- Huh
- Huhn
- Ei muuten tohi

VIIKON TARJOUS

Yömelonta 23km

Koe raikastava elämys
yötönnän yön melonta-
retkellä...
[Lue lisää](#)

KYSY
HINTAA!

AJANKOHTAISTA

- > Melontareittimme uudistuvat
- > Chilling -seikkailu-paketti saapui
- > Kesäaktiiviteettien uudet hinnat julkaistu

TEE VARAUS

Tiedatko jo mitä haluat?
Tee seikkailuvarausesi
suoraan netissä!

[VARAA HETI](#)

OTA YHTEYTÄ

TEE VARAUS

USEIN KYSYTTÄ

KESÄAKTIVIITEITIT

- > Koskenlasku
- > Melonta
- > Maastopyöräily
- > Montijäsfararit
- > Vaelukset

TALVIAKTIVIITEITIT

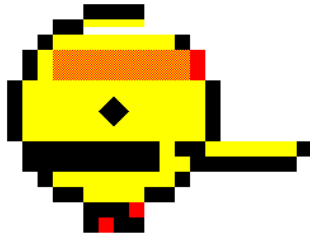
- > Keikkasafarit
- > Porosafarit
- > Husky safarit
- > Jäägolf
- > Jääkieppely

ACID2- JA ACID3-TESTITULOKSET

Liite 2

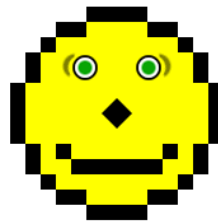
Kaikki testit ajettiin Windows 7 ympäristössä, lukuunottamatta Internet Explorer 6:n testejä, jotka ajettiin Windows XP ympäristössä käyttäen Microsoft Virtual PC:tä.

Hello World!



Firefox2

Hello World!

Firefox 3.6.3
Safari 4.05
Opera 10.53

Hello World!



Internet Explorer 8

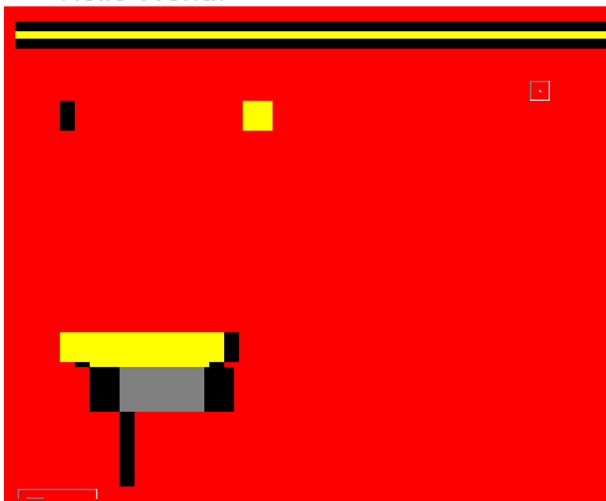
Acid2-testitulokset eri selaimilla. Mozilla Firefox 3.6.3, Safari 4.05 ja Opera 10.52 selaimet läpäisivät testin. Mozilla Firefox 2.0 ja Internet Explorer 8 eivät osanneet näyttää HTML:llä ja CSS:llä rakennettua hymiötä oikein.

Hello World!



Kuvankaappaus Internet Explorer 7 selaimesta

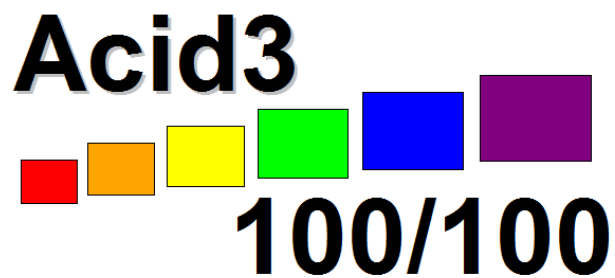
Hello World!



Kuvankaappaus Internet Explorer 6 selaimesta

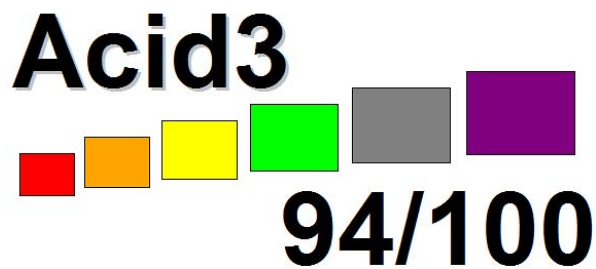
Internet Explorerin versioilla 6 ja 7 oli pahimmat ongelmat kuvan näyttämisen kanssa.

Acid3-testitulokset eri selaimilla. Ainoastaan Opera 10.5 ja Safari 4 läpäisivät Acid3 testin moitteetta. Mozilla Firefox 3.6 oli myös lähellä läpäisyä ja sai testissä 94/100 pistettä. Microsoftin viimeisin selain Internet Explorer 8 läpäisi testin vain 20 prosenttisesti.



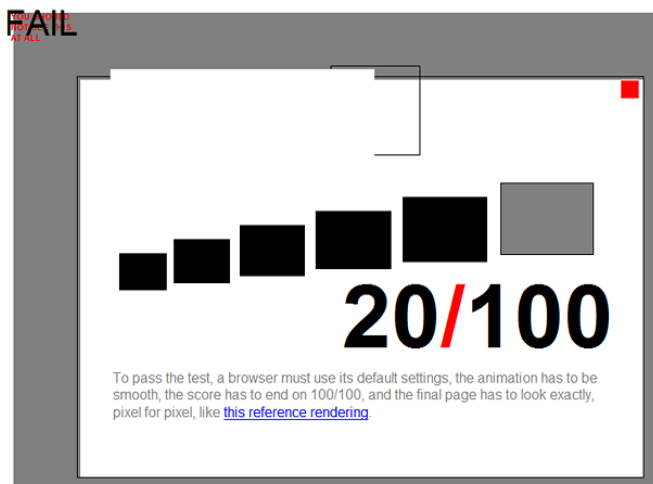
To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like [this reference rendering](#).

Opera 10.5 ja Safari 4.05 selainten kuvankaappaus Acid3-testistä.

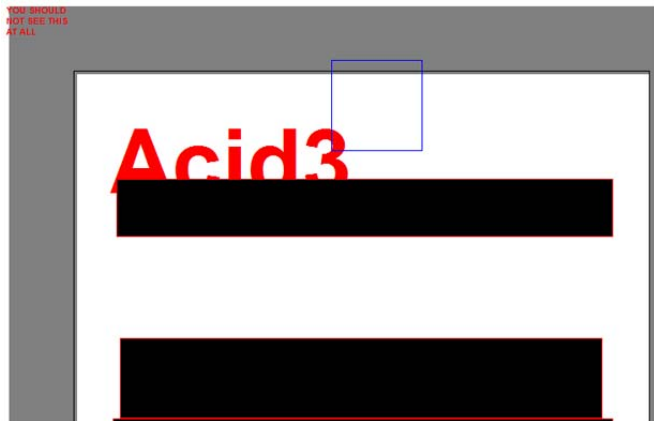


To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like [this reference rendering](#).

Mozilla Firefox 3.6 selaimen kuvankaappaus Acid3-testistä.

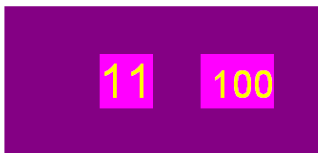


Internet Explorer 8 selaimen kuvankaappaus Acid3-testistä.



Internet Explorer 7 selaimen kuvankaappaus Acid3-testistä.

Acid3

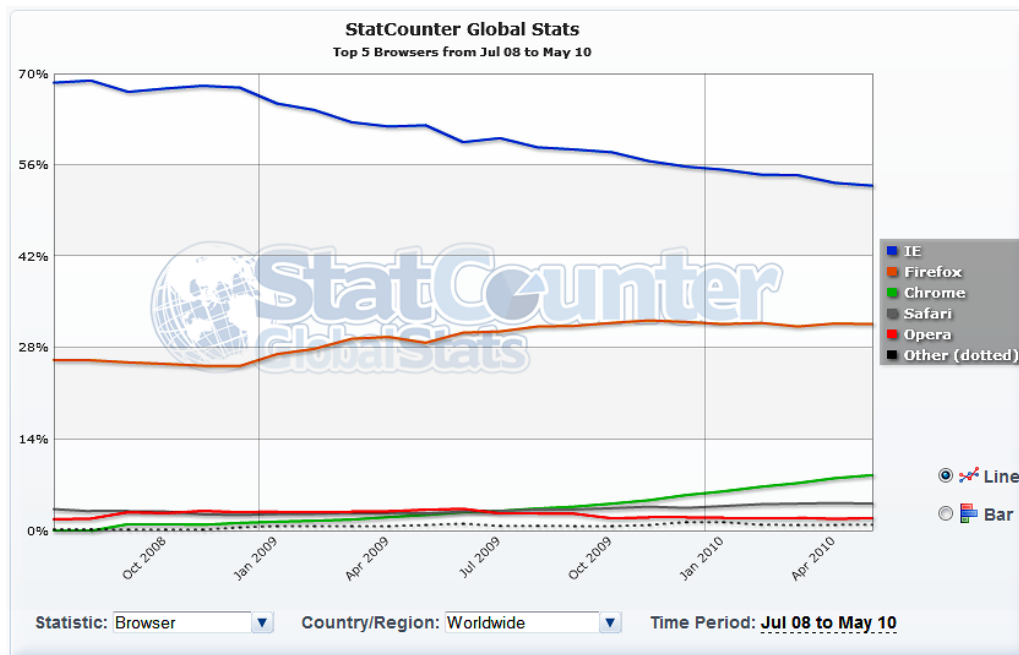


To pass the test, a browser must use its default settings, the animation has to be smooth, the score has to end on 100/100, and the final page has to look exactly, pixel for pixel, like [this](#)

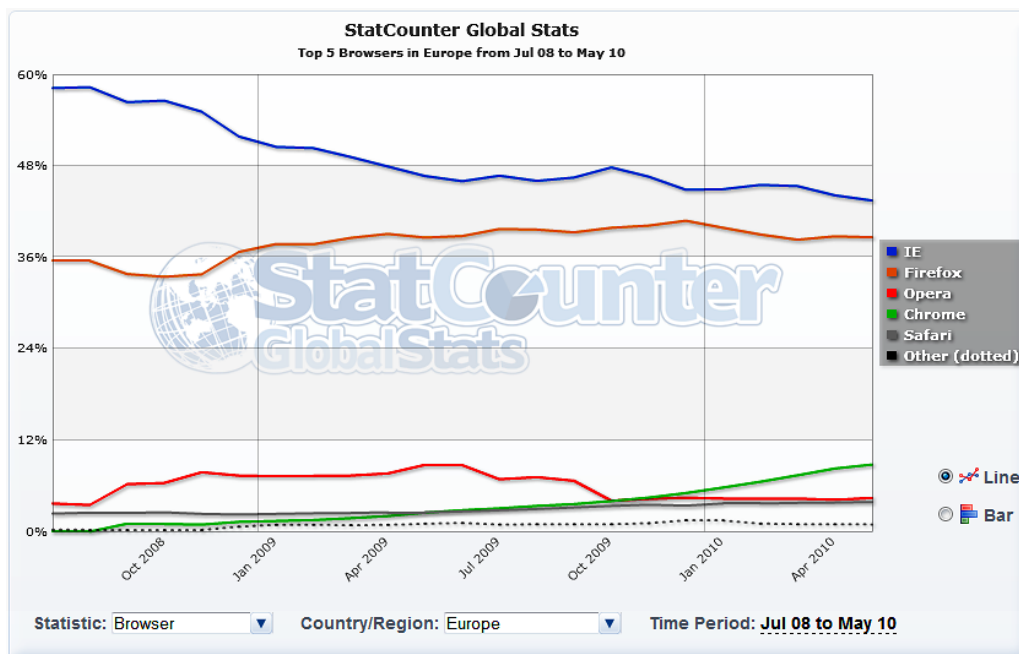
Internet Explorer 6 selaimen kuvankaappaus Acid3-testistä.

SELAINUTILASTOT GRAAFISESTI ESITETTYNÄ

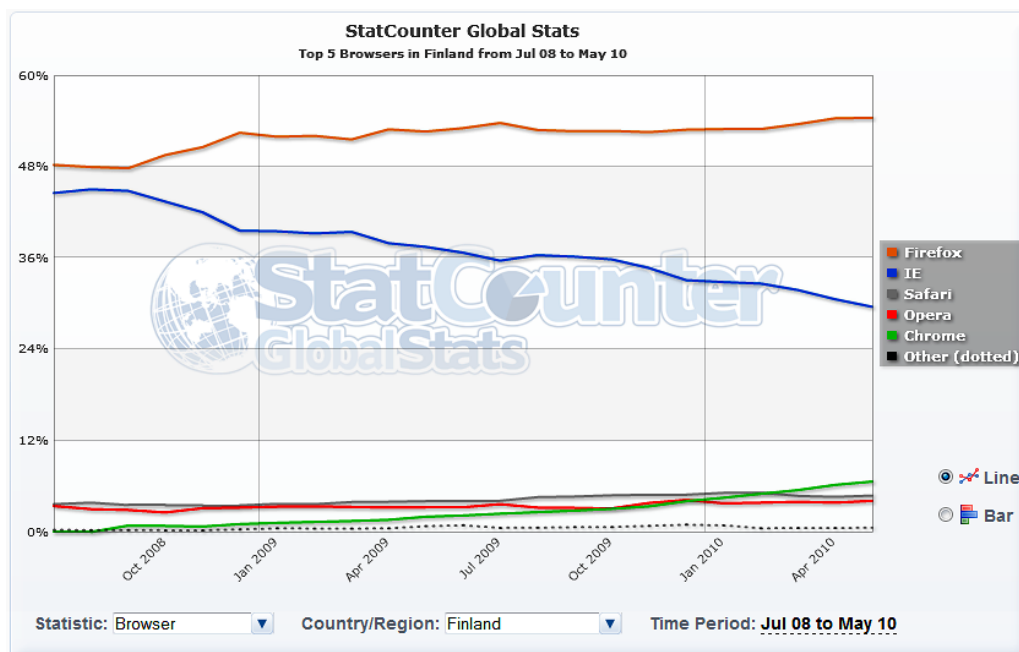
Liite 3



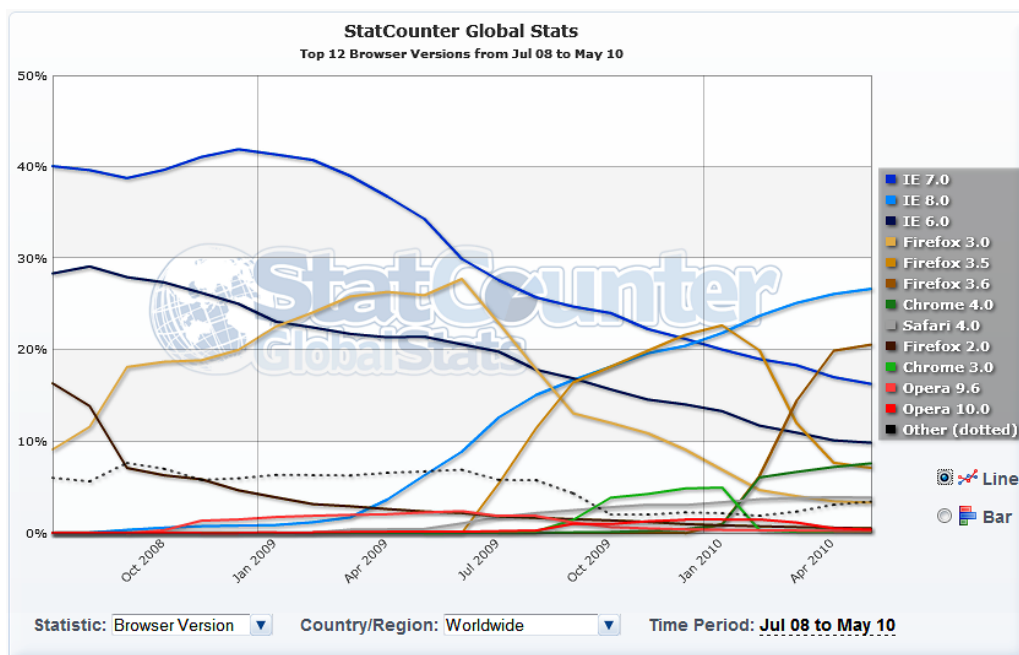
Selaintilastot maailmanlaajuisesti Heinäkuu 2008 – Huhtikuu 2010 väliseltä ajalta. Vaakatasolla on aikajana ja pystypalkista on nähtävissä selaimen markkinaosuus prosentteina.



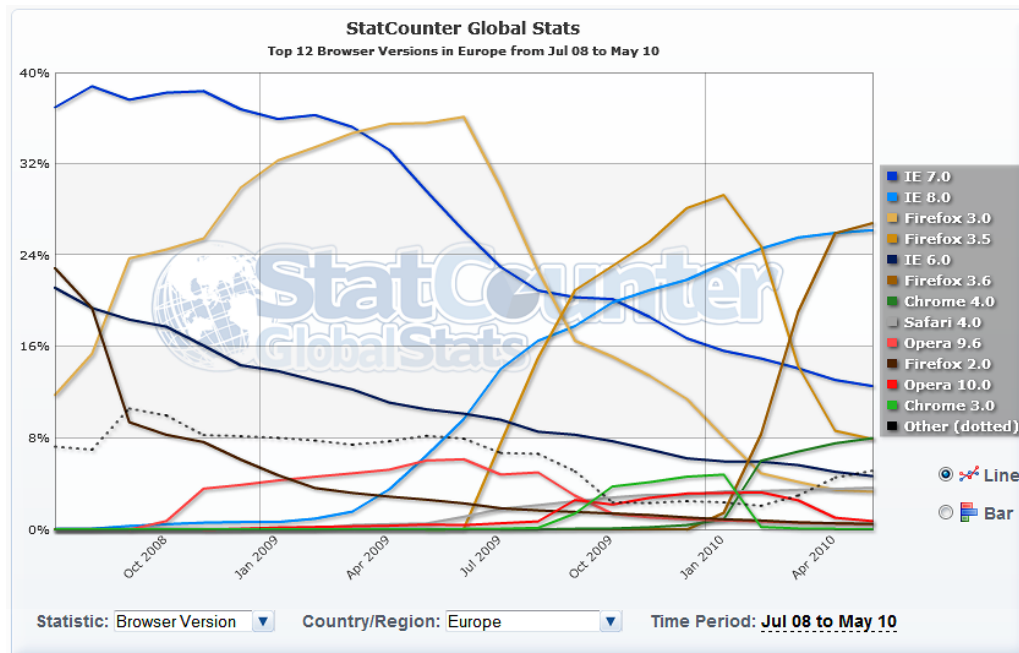
Selaintilastot Euroopan alueella Heinäkuu 2008 – Huhtikuu 2010 väliseltä ajalta. Vaakatasolla on aikajana ja pystypalkista on nähtävissä selaimen markkinaosuus prosentteina.



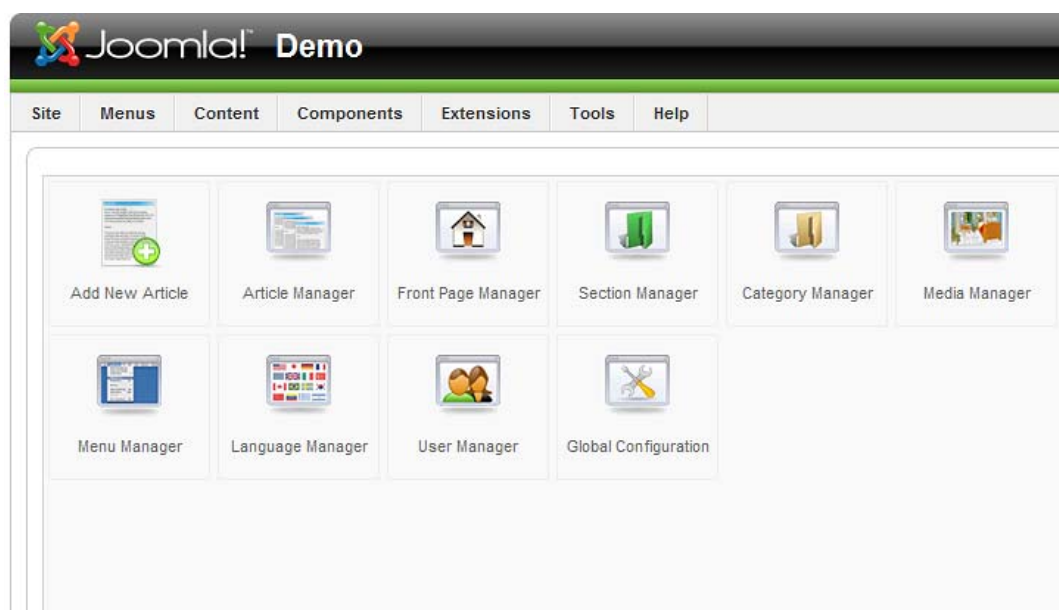
Selaintilastot Suomessa kerätystä datasta Heinäkuu 2008 – Huhtikuu 2010 väliseltä ajalta. Vaakatasolla on aikajana ja pystypalkista on nähtävissä selaimen markkinaosuus prosentteina.



Selainten versiokehitys maailmanlaajuisesti Heinäkuu 2008 – Huhtikuu 2010 väliseltä ajalta. Vaakatasolla on aikajana ja pystypalkista on nähtävissä selaimen markkinaosuus prosentteina. Kaaviosta huomataan, että kehitys on aivan oikean suuntainen, sillä vuoden sisällä vanhempien selainten osuus markkinoilla on selvästi pienentynyt ja uusien vastaavasti kasvanut, kun selaimia on päivitetty.



Selainten versiokehitys Euroopan alueella Heinäkuu 2008 – Huhtikuu 2010 väliseltä ajalta. Vaakatasolla on aikajana ja pystypalkista on nähtävissä selaimen markkinaosuus prosentteina. Kaaviosta huomataan, että kehitys on rajumpaa Euroopan alueella kuin maailmanlaajuisesti. Tästä voisi päätellä, että Euroopassa pidetään selainversiot paremmin ajantasalla kuin mitä muualla maailmassa.



Kuvankaappaus Joomla!n ylläpitopaneelistä.

Show descriptions

Content management

Manage your site's content.

- [Comments](#)
- [Content](#)
- [Content types](#)
- [Post settings](#)
- [RSS publishing](#)
- [Taxonomy](#)

User management

Manage your site's users, groups and access to site features.

- [Access rules](#)
- [Permissions](#)
- [Roles](#)
- [User settings](#)
- [Users](#)

Reports

View reports from system logs and other status information.

- [Recent log entries](#)
- [Testisivu](#)
- [Top 'access denied' errors](#)
- [Top 'page not found' errors](#)
- [Status report](#)

Site building

Control how your site looks and feels.

- [Blocks](#)
- [Menus](#)
- [Modules](#)
- [Themes](#)

Site configuration

Adjust basic site configuration options.

- [Actions](#)
- [Administration theme](#)
- [Clean URLs](#)
- [Date and time](#)
- [Error reporting](#)
- [File system](#)
- [Image toolkit](#)
- [Input formats](#)
- [Logging and alerts](#)
- [Performance](#)
- [Site information](#)
- [Site maintenance](#)

Kuvankaappaus Drupalin ylläpitopaneelistä.

[illegible]